한국시뮬레이션학회 논문지
Vol. 15, No. 3, pp. 115-124 (2006. 9)

# 유사성 비교를 통한 RDB의 참조 무결성 관계 추출 알고리즘

김장원[1] · 정동원[2] · 김진형[1] · 백두권[1†]

# An Algorithm for Referential Integrity Relations Extraction using Similarity Comparison of RDB

**Jangwon Kim · Dongwon Jeong · Jinhyung Kim · Doo-Kwon Baik**

ABSTRACT

XML is rapidly becoming technologies for information exchange and representation. It causes many research issues such as semantic modeling methods, security, conversion for interoperability with other models, and so on. Especially, the most important issue for its practical application is how to achieve the interoperability between XML model and relational model. Until now, many suggestions have been proposed to achieve it. However, several problems still remain. Most of all, the exiting methods do not consider implicit referential integrity relations, and it causes incorrect data delivery. One method to do this has been proposed with the restriction where one semantic is defined as only one same name in a given database. In real database world, this restriction cannot provide the application and extensibility. This paper proposes a noble conversion (RDB-to-XML) algorithm based on the similarity checking technique. The key point of our method is how to find implicit referential integrity relations between different field names presenting one same semantic. To resolve it, we define an enhanced implicity referential integrity relations extraction algorithm based on a widely used ontology, WordNet. The proposed conversion algorithm is more practical than the previous-similar approach.

**Key words** : RDB, XML, WordNet, Referential Integirty, Similarity

요 약

XML은 정보 교환과 표현을 위해 빠르게 발전해 오고 있는 기술이다. XML을 통한 시멘틱 모델링 방법론, 보안, 다른 모델들과의 상호 운용성을 위한 변환과 같은 많은 연구들이 이슈화 되었다. 특히, 실질적인 응용분야의 가장 중요한 이슈는 XML모델과 관계형 모델들과의 상호 운용성을 어떻게 획득하는 것인가이다. 지금까지 상호 운용성을 위해 많은 방법들에 제기되어 왔다. 하지만, 여전히 몇 가지 문제점이 있다. 대부분의 기존의 방법들은 묵시적인 참조 무결성 관계를 고려하지 않기 때문에, 부정확한 데이터 전달이 야기된다. 데이터베이스에서 하나의 의미가 정의 될 때 오직 하나의 이름만 가진다는 제약조건하에서 위의 문제를 해결하기 위한 한 가지 방법이 제안되었다. 하지만, 실제 데이터베이스에서 응용과 확장을 위해서 이 제약사항을 적용할 수는 없다. 그래서 이 논문에서는 유사성 검사 기법을 기반으로 한 RDB-to-XML 변환 알고리즘을 제안한다. 이 방법의 핵심은 하나의 같은 의미에 대해 다른 이름으로 표현되는 속성들 간의 묵시적인 참조 무결성 관계를 찾아내는 것이다. 이것을 해결하기 위하여 널리 이용되는 온톨로지인 WordNet을 기반으로 하여 향상된 묵시적 참조 무결성 관계를 추출하는 알고리즘을 정의하였다. 제안된 변환 알고리즘은 이전의 유사한 접근 방법 보다 더욱 실질적이다.

**주요어** : 관계형데이터베이스, XML, 워드넷, 참조 무결성, 유사성

* 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받았음.
2006년 6월 19일 접수, 2006년 9월 7일 채택
[1] 고려대학교 컴퓨터학과
[2] 국립군산대학교 정보통계학과
주 저 자 : 김장원
교신저자 : 백두권
E-mail; jwkim@software.korea.ac.kr, djeong@kusan.ac.kr, koolmania@software.korea.ac.kr, Baik@software.korea.ac.kr

## 1. 서 론

XML의 등장으로 인해 다양한 데이터들을 XML형태로 표현하는 많은 연구들이 진행되고 있다[1]. 대표적인 연구로는 관계형 데이터베이스에 저장되어 있는 정보를 XML문서로 변환하기 위한 연구로써[2] 이를 위해 다양한 변환 방법(알고리즘)들이 제안되었다[3,4]. 이러한 방법 중 하나인 Constraints-based Translation(CoT) 알고리즘은 변환 시 관계형 데이터베이스 스키마를 통해 추출 가능한 명시적 참조 무결성 관계만을 XML문서에 반영 한다[3]. 그래서 관계형 데이터베이스 스키마에 묵시적 참조 무결성 관계가 존재할 경우 XML 문서에 이를 정확하게 반영하지 못하며 이는 부정확한 정보 교환을 발생하게 함으로써 향후 이상 현상 등을 초래할 수 있다. 이러한 문제를 해결하기 위해 Values Pattern-based RDB-to-XML Translation (VP-T) 알고리즘은 데이터베이스 내 값들에 대한 패턴들을 분석하여 명시적으로 정의되지 않은 참조 무결성 관계를 자동으로 추출함으로써 위와 같은 문제를 해결한다. 그러나 VP-T알고리즘은 각 테이블에 구성된 컬럼의 값을 이용하여 비교하기 때문에 컬럼의 이름이 같은 의미를 가지지만 다른 용어를 사용하였을 경우 이들 컬럼들 간에 가질 수 있는 묵시적 참조 무결성 관계를 추출하지 못한다. 또한, 컬럼의 이름은 완전한 형태의 단어이고 하나의 의미를 표현하기 위해 동일한 컬럼 이름(필드 명)을 사용한다고 가정하고 있다. 이는 실질적인 응용에 많은 제약이 따른다. 따라서 이러한 문제점을 해결하기 위해 같은 의미를 가지지만 다른 용어가 사용된 컬럼들의 묵시적 참조 무결성 관계 정보를 추출하기 위한 연구가 필요하다.

이 논문에서는 위의 문제를 해결하기 위하여 현재 널리 사용되고 있는 온톨로지 기반의 워드넷을 이용하여 묵시적 참조 무결성 관계 정보를 추출하는 알고리즘을 제안한다. 이 논문은 다음과 같이 구성된다. 제 2장에서는 관련 연구로써 기존에 사용되었던 변환 방법과 워드넷에 대하여 서술한다. 제 3장에서는 관계형 데이터베이스의 정보를 XML 문서로 변환하기 위한 변환 모델을 정의한다. 제 4장에서는 묵시적 참조 무결성 관계 정보 추출을 위한 알고리즘을 제안한다. 제 5장에서는 묵시적 참조 무결성 관계를 추출하기 위한 프로세스 및 시스템의 각 구성 요소를 서술하고, 제 6장에서는 결론 및 향후 연구에 대해 제시한다.

## 2. 관련 연구

관계형 데이터베이스 스키마를 XML 문서로 변환하기 위한 다양한 방법들이 제안되었다. 2장에서는 그 중 가장 대표적인 CoT 알고리즘과VP-T 알고리즘의 특징에 대하여 기술한다.

### 2.1 CoT

CoT 알고리즘은 테이블, 컬럼 등과 같은 관계형 데이터베이스의 구조적인 부분 뿐 아니라 테이블의 제약 조건, 참조 무결성 등의 관계형 데이터베이스에서 가지고 있는 의미적인 부분까지 변환한다. CoT는 외부 키 제약조건을 고려하여 제약 조건이나 참조 무결성 관계 정보를 이용하여 서로 관계를 가지는 테이블을 계층적인 구조로 표현하여 관계형 데이터베이스의 정보들을 XML 스키마로 변환시킨다. 하지만, CoT알고리즘은 변환 시 명시적으로 정의된 참조 무결성 관계만을 고려한다. 즉, 변환 시 관계형 데이터베이스에 존재하는 묵시적 참조 무결성 관계 정보들을 XML문서에 반영할 수 없기 때문에 많은 정보 손실을 야기한다.

### 2.2 Values Pattern-based RDB-to-XML Translation(VP-T)

VP-T알고리즘은 앞 절에서 소개 되었던 CoT 알고리즘을 보완한 것이다. 관계형 데이터베이스 스키마를 XML 문서로 변환 시 CoT알고리즘으로는 반영할 수 없었던 묵시적 참조 무결성 관계 정보를 고려한다. VP-T알고리즘은 관계형 데이터베이스에서 제공하는 메타 데이터 정보를 이용하여 테이블 및 컬럼들에 대한 정보를 획득한다. 이런 정보들을 이용하여 대상 컬럼을 추출하고 동일한 이름을 사용하고 있는 컬럼들의 데이터들을 비교한다. 비교를 통하여 비교 대상의 컬럼의 값이 다른 컬럼에 여러번 반복하여 나타나게 된다면 이들 컬럼은 참조 무결성 관계를 가진다고 판단한다. 이런 방법을 통하여 컬럼들의 묵시적 참조 무결성 관계 정보를 찾아낸다. 하지만, VP-T 알고리즘에서는 묵시적 참조 무결성 관계 정보를 가지는 컬럼들은 이름이 동일하다고 가정한다. 그러므로 해당 컬럼들의 이름이 다를 경우 컬럼들의 값을 비교를 할 수 없기 때문에 묵시적 참조 무결성 관계를 가지고 있는지 판단 할 수 없다. 이런 조건 때문에 컬럼들이 동일한 의미를 가지는 경우 여러 테이블에서 무조건적으로 동일한 컬럼 이름으로만 정의해야 한다는 제약성은 그 실용성이나 활용성에 측면에서 매우 제한적이다. 또한 대상 컬럼들 간

의 이름이 동일하여도 데이터베이스에 존재하는 실제 데이터들이 1:N 관계가 아닌 1:1 또는 1:0 관계를 보일 경우 묵시적 참조 무결성 관계 정보를 추출할 수 없다.

### 2.3 워드넷(WordNet)

워드넷은 인간의 어휘 지식에 대한 심리언어학 연구의 성과를 토대로 1985년 프린스톤 대학의 인지과학 연구실이 구축해온 언어의 어휘 데이터베이스이다[5]. 워드넷의 주된 특징은 단어 형태가 아닌 단어의 의미를 기반으로 한 어휘 사전을 만든 것이다. 따라서 워드넷은 어휘의 저장 형태가 어휘의 의미를 이용한 계층적 구조로 이루어져 있다. WordNet에서 명사는 25개 범주로 나뉘어 계층구조를 이루고 있는데 계층의 깊이는 12단계를 넘지 않는다[6]. 25개의 범주와 계층구조를 이용하여 어휘간의 거리를 파악할 수 있으며, WordNet의 분류범주는 배타적인 것이 아니어서 서로 다른 범주에 속한 개념들 사이에도 상하의 관계 이외의 다른 관계, 심지어는 분의관계까지도 설정할 수 있다. 따라서 비슷한 의미를 지니는 어휘를 입력하게 되면 계층구조를 이용해서 서로 간에 얼마만큼 밀접한 관계를 가지고 있는지 알 수 있다.

그래서 이 논문에서는 컬럼의 이름에 대한 유사성 비교를 위해 워드넷을 이용한다.

## 3. 변환 모델

이 장에서는 관계형 데이터베이스의 스키마를 XML 문서로 변환하기 위한 변환 모델들을 정의한다. 변환 모델로는 초기 입력으로 초기 관계형 스키마 모델, 중간 결과인 워드넷을 이용한 관계형 스키마 모델이 있다. 초기 관계형 스키마 모델과 XML 스키마 모델은 기존의 많은 연구에서 다루어졌다[3,7,8]. 따라서 기본적인 모델들의 정의를 위한 표기 등은 기본 연구내용을 기반으로 한다. 또한, 워드넷을 이용하여 묵시적 참조 무결성 관계 정보를 추출한 것들에 대한 표현을 하기 위해서 기존 모델을 확장한다.

### 3.1 초기 관계형 스키마 모델

관계형 데이터베이스에서 스키마는 테이블 명, 컬럼명, 컬럼 타입, 제약사항들로 구성되며, 제약사항은 Not Null, 외래키, Unique등이 있다. 변환 시 이들을 고려하여 XML 문서에 반영되어야 한다.

초기 관계형 스키마 모델은 다음과 같은 5개의 튜플로 구성된다.

[정의 1] $R_{input}$ = (T, C, P, $RI_{exp}$, K)

- T는 테이블의 유한 집합을 표현한다.
- C는 각 테이블에 존재하는 컬럼의 집합을 표현하기 위한 함수
- P는 각 컬럼의 속성을 표현하기 위한 함수
  - t는 정수형, 문자열과 같은 컬럼의 데이터 형태를 표현
  - u는 컬럼 값의 유일성을 표현
  - n은 컬럼 값의 Null 가능 여부를 표현
- $RI_{exp}$는 명시적인 참조 무결성과 연관된 컬럼의 쌍을 표현한다.
- K는 주 키 정보를 표현하기 위한 함수

### 3.2 워드넷을 이용한 관계형 스키마 모델

워드넷을 이용한 관계형 스키마 모델은 8개의 튜플로 표기 할 수 있다.

[정의 2] $R_{wordnet}$ = (T, C, P, K, $RI_{exp}$, W, F, $RI_{imp}$)

- T, C, P, K, $RI_{exp}$ 는 초기 관계형 스키마 모델과 동일한 요소이다.
- $RI_{imp}$는 묵시적 참조 무결성 관계를 가지는 컬럼들의 쌍을 표현한다.
- W는 워드넷으로 부터 얻어온 정보의 집합을 표현한다.
- F는 유사성 체크를 하기 위한 함수

위에서 정의된 모델에 초기 관계형 스키마 모델을 적용하면 다음과 같이 표현할 수 있다.

$$R_{wordnet} = (T, C, P, K, RI_{exp}, W, F, RI_{imp})$$
$$= (R_{input}, W, F, RI_{imp})$$

## 4. RDB to XML 변환

관계형 데이터베이스의 메타데이터에는 각 컬럼들의 속성 정보를 가지고 있으며, 그 중 하나로 외부 키 제약조건이 있다. 따라서 메타데이터를 이용하면 관계형 데이터베이스의 외부 키 제약 조건에 대한 속성 정보를 고려하여 XML문서로 변환할 수 있다. 변환 시 관계형 데이터베이스에서 가지고 있는 정보들을 고려해서, 관계형 데이터베이스에 있는 묵시적 참조 무결성 관계 정보가 XML 문서에 반영되도록 해야 한다. 이 논문에서는 묵시적 참조 무결성 관계 정보를 추출하기 위하여 다음과 같은 알고리즘을 제안한다.

### 4.1 알고리즘

[용어 설명]

- i, j, length는 정수형 변수
- T[ ]는 테이블의 모든 컬럼 이름 집합
- $T_{new}$[ ]는 LLD에 저장되어 있지 않고 새롭게 검색을 하는 컬럼 명들의 집합
- $T_{idx}$[ ]는 컬럼 이름을 워드넷을 통하여 검색할 때 대상 컬럼의 이름이 워드넷에 들어 있을 경우 계층구조로 이루어져 있는 워드넷에서 해당 컬럼의 이름이 계층구조상의 어떤 레벨에 위치해 있는지 알 수 있는 인덱스 값을 저장
- $T_{list}$[ ], $T_{morph}$[ ], $T_{tree}$[ ]는 입력 된 어휘에 대하여 워드넷으로부터 얻어온 결과 집합
- $T_{can}$[ ][ ]는 워드넷으로부터 얻어온 모든 결과들의 집합
- TLLD[ ][ ]는 검색하고자 하는 어휘가 Local Lexical Dictionary에 저장되어 있을 경우 관련 정보를 저장
- Trim_Lang은 컬럼들에서 비자연어 적인 어휘를 제거하기 위한 함수
- Check_LLD는 LLD에 검색을 원하는 어휘들의 정보가 사전에 들어가 있는지 체크하는 함수
- Wn_Input()은 워드넷을 이용하여 동일한 의미를 가지지만 이질적인 단어로 쓰인 것들을 찾기 위한 함수
- lookupIndexWord()는 입력된 특정 컬럼의 이름이 워드넷 계층구조 상에서 어디에 있는지 알기 위한 함수
- demonstrateListOperation()는 인덱스를 이용하여 유사성을 가지는 어휘를 검색 함수
- Check_Similarity()는 유사성을 가지는 어휘들과 T[ ]에 들어 있는 컬럼들과 비교를 하여 유사성을 가지는지 검사하는 함수

표 1의 알고리즘은 다음과 같다. 관계형 데이터베이스에 있는 메타데이터 정보를 가져 온다. 가져온 정보들로부터 테이블에 있는 컬럼의 이름을 추출한다. 그리고 Trim_Lang 함수에 컬럼들의 이름을 입력 한다. Trim_Lang 함수는 다음과 같은 역할을 한다.

- 컬럼의 이름을 보다 명확하고 직관적으로 알기 위해 붙여진 접미사, 또는 접두어 들이 특정 컬럼(주키)의 이름에 중복되어 있을 경우 잘라낸다.

만약 워드넷을 이용할 때 다른 단어들이 붙어 있는 경우 이것을 제거해야 해당 단어와 의미가 같은 어휘를 추출할 수 있다. 이 논문에서 제안하는 알고리즘에는 기존 상용 데이터베이스인 Oracle의 SQL Shared Pool 영역과 유사한 역할을 하기위한 목적으로 Local Lexical Dictionary(LLD)를 만들어 워드넷을 이용하여 검색한 단어들의 결과 집합들을 저장함으로써 추후 동일한 단어를 검색 시 시간을 줄일 수 있다.

다음 단계(Line 8)로 워드넷에 LLD에는 없는 새로운 컬럼 명들을 입력한다. 이때 지정한 경로에 워드넷이 있는지 확인하고, 워드넷 인스턴스를 생성한다. 생성된 인스턴스를 이용하여 컬럼 이름들에 대한 인덱스를 얻어 온다.

**표 1.** 제안 알고리즘

```
1. Initialize :
   i=0, j=0, relation=0, length=0, T[ ]=0, Tnew[ ]=0,
   Temp[ ]=0, Tidx[ ]=0, Tlist[ ]=0, Tmorph[ ]=0, Ttree[ ]=0,
   Tcan[ ][ ]=0, TLLD[ ][ ]=0
2. Connect_to_DB()
3. Temp[c] = getMetadata(Database_Name)
4. if (is there a non-natural-lang) {
            T[ ] = Trim_Lang(Temp[ ])}
5. else T[ ] = Temp[ ]
6. if (not boolean Check_LLD(T[ ])) {
7.   Tnew[ ] = T[ ]
8.    Wn_Input(Tnew[ ]) {
9.    length = Length(Tnew[ ])
10.    JWNL.initialize()
11.    Set_Dictinoary_WordNet(info, version, location)
12.    Dictionary.getInstance() {
          Create Wordnet Instance }
13.    lookupIndexWord(Tnew[ ]) {
          for i = 0 to length { Tidx[i] =
       Dictionary.getInstance().getIndexWord(Tnew[i])}
14.    go(Tidx[ ]) {
          for i = 0 to length {
15.       Tlist[ ] = demonstrateListOperation(Tidx[i])
16.       Tcan[i][ ] = Tidx[i] + Tlist[ ] }}}
17.   Save_To_LLD(Tcan[ ][ ])}
18. else { return TLLD[ ][ ] }
19. Check_Similarity(T[ ], Tcan[ ][ ], TLLD[ ][ ]) {
20.         Timp[ ][ ] = Add(Tcan[ ][ ], TLLD[ ][ ])
            for i to length {
              for j to Length(Timp[i][ ]) {
                if (T[i] == Timp[i][j])
                   print these are one pair }
21    relation = Check_asymmetric_Relation(T[ ], Timp[ ])
22.   return (T[i], Timp[i]), relation}}
```

Student

| StudentID | Sname | Pid | Cname |
|---|---|---|---|
| s01 | Alice | p01 | Database |
| s02 | Tim | p02 | HCI |
| s03 | Neo | p02 | Algorithm |
| s04 | Smith | p03 | Simulation |
| s05 | Alba | p04 | Database |

Class

| Cname | Place | Time |
|---|---|---|
| Database | 101 | #1 |
| HCI | 103 | #5 |
| Algorithm | 121 | #2 |
| Simulation | 124 | #6 |

Room

| OfficeID | Pid | Rname |
|---|---|---|
| 101 | p01 | DB Lab |
| 103 | p04 | HCI Lab |
| 108 | p03 | Mobile Lab |
| 121 | p02 | NLP Lab |
| 124 | p02 | Lecture2 |
| 209 | p05 | OS Lab |
| 217 | p01 | Lecture1 |
| 213 | p02 | Lecture3 |
| 225 | p04 | Lecture4 |

Professor

| Pid | Pname | Office | AssistantID |
|---|---|---|---|
| p01 | Dr.Kim | 217 | s01 |
| p02 | Dr.Park | 213 | s02 |
| p03 | Dr.Song | 108 | s04 |
| p04 | Dr.Cho | 225 | s03 |
| p05 | Dr.Ryu | 209 | s05 |

그림 1. 관계형 데이터베이스 테이블

그리고 인덱스를 이용하여 워드넷의 어휘사전에 의미가 같지만 이질적인 형태의 단어들을 추출한다. 이 결과는 LLD에 저장되고, 처음 입력된 컬럼들과 워드넷을 이용하여 나온 유사 어휘 집합들을 비교하여 유사성을 가진 컬럼들은 각각의 쌍으로 묶는다. 이렇게 쌍으로 묶여진 어휘들을 묵시적 참조 무결성 관계를 가지는 후보군으로 지정한다.

### 4.2 변환 예제

이 절에서는 관계형 스키마 모델의 XML 스키마 모델로의 변환 과정을 보여준다. 데이터베이스에는 Student (Sid, Sname, Pid, Cname), Class(Cname, Place, Time), Professor(Pid, Pname, Office, AssistantID), Room(Rid, Managerid, Rname) 의 4개의 테이블과 14개의 컬럼들을 가지고 있다. 학생들은 지도교수가 한명 씩 있으며, 학생은 한 과목 이상을 수강한다. 강의는 고유한 수업명과 장소, 시간을 가진다. 교수는 유일한 교수번호가 있으며, 자신의 연구실을 가지고 있다. 또한, 수업 조교를 한 명씩 두고 있다. 수업 조교는 학생들 중의 한명으로 학생은 여러 번 조교를 할 수 있다. 방에는 유일한 방 번호를 가지며 강의실과 교수실 모두를 가지고 있다. 각 방은 관리 교수가 한 명 씩 있으며 교수는 여러 개의 방을 관리 할 수 있다. 방은 용도에 따라 각각의 이름을 가진다.

관계형 데이터 정보를 XML 로 변환하기 위해 3.1 의 [정의 1]을 이용하여 표 2에 표현하였다.

표 2의 초기 관계형 스키마 모델은 묵시적 참조 무결성 관계를 명시적으로 표현하지 않는다. 묵시적 참조 무결성 관계를 추출하기 위해 이 논문에서 제안한 알고리즘을 샘플 데이터에 적용하면 다음과 같다. 우선 관계형 스키마

표 2. 초기 관계형 스키마 모델

**테이블 집합 T**

T = {Student, Professor, Class, Room}

**컬럼 집합 C(table_name)**

C(Student) = {StudentID, Sname, Pid, Cname}
C(Class) = {Cname, Place, Time}
C(Professor) = {Pid, Pname, Office, AssistantID}
C(Room) = {OfficeID, ManagerID, Rname}

**주키 컬럼 K(key_column)**

K(Student) = {StudentID}
K(Class) = {Cname}
K(Professor) = {Pid}
K(Room) = {OfficeID}

**컬럼들의 속성 P(column_name)**

P(StudentID) = {string, u, !n}
P(Sname) = {string, u, !n}
P(Pid) = {string, u, !n}
P(Cname) = {string, u, !n}
P(Cname) = {string, u, !n}
P(Place) = {string, u, !n}
P(Time) = {string, u, n}
P(Pid) = {string, u, !n}
P(Pname) = {string, u, !n}
P(Office) = {string, u, !n}
P(AssistantID) = {string, u, !n}
P(OfficeID) = {string, u, !n}
P(ManagerID) = {string, u, !n}
P(Rname) = {string, u, !n}

에서 제공하는 메타데이터를 이용하여 컬럼들의 정보를 가져온다. 만약 컬럼 이름에 Sid, Pid, Sname, Cname 과 같은 비자연어적인 단어가 있을 경우 의미가 같은 단어 집합을 찾지 못하는 경우가 발생할 수 있다. 이 와 같은 경우 VP-T 방법을 이용하면 묵시적 참고 무결성 관계를 추출해 낼 수 있다. 또한 논리 데이터베이스의 표현을 위하여 주 키 속성을 가지는 컬럼들의 이름에서 특정 어휘들(AssistantID, StudentID → ID)을 사용한 경우 이것을 잘라내 워드넷에서 명확한 한 개의 단어로써 그것이 가지는 의미를 정확하게 추출하여 이 논문에서 워드넷을 이용하여 얻고자 한 목적인 의미가 같지만 이질적인 단어를 찾아 낼 수 있도록 한다. 아래의 표 3은 위에서 설명한 과정을 거친 대상 컬럼들의 이름을 워드넷에 입력하여 추출

**표 3.** 유사 어휘 추출

| |
|---|
| W(Office) = {spot, billet, place, ... , situation} |
| W(Place) = {topographic point, place, spot} |
| W(Assistant) = {helper, help, supporter} |
| W(Student) = {pupil, educatee} |
| W(Time) = {time} |

**표 4.** 유사 어휘간의 관계

| | Place | Office | Assistant | Time | Student |
|---|---|---|---|---|---|
| Place | 0,0 | 3,9 | 3,9 | Null | 3,9 |
| Office | 3,5 | 0,0 | 5,10 | Null | 5,10 |
| Assistant | 6,9 | 5,10 | 0,0 | Null | 2,4 |
| Time | Null | Null | Null | 0,0 | Null |
| Student | 6,9 | 5,10 | 2,4 | Null | 0,0 |

한 어휘들의 집합이다.

표 3의 결과를 이용하여 Office와 Place가 서로 연관 관계가 있음을 알 수 있다. 하지만 최초에 제시한 논리모델에서 조교는 학생이 한다고 가정하였기 때문에 이와 관계된 묵시적 참조 무결성 정보를 위의 표 3의 결과에서 얻어야 하지만 서로 유사한 의미를 가지고 있는 단어가 아니기 때문에 원하는 정보를 얻을 수가 없다.

위에서 발생한 문제점을 해결하기 위해 워드넷에 있는 단어 간 계층 구조를 이용하여 단어가 가지고 있는 인덱스 값을 찾아 비교함으로써 묵시적 참조 무결성 관계 정보를 얻어 낼 수 있다. 표 4는 각 단어의 관계를 계층 구조를 통하여 알아 본 것이다. 표 안의 첫 번 째 숫자는 인덱스 거리를 의미하며, 두 번째 숫자는 깊이를 나타낸다. 위의 표에 Student와 Assistant의 관계는(2, 4)를 나타내고 있다. Student는 자기 자신(Student)과의 인덱스 거리가 0이고 트리의 깊이가 0인 것을 확인 할 수 있다. 인덱스 거리가 0에 가까울수록 비교 대상 단어와의 관계성은 높아진다. 즉, 각 단어의 최상의 synset이 서로 연관되어 있다는 것을 알 수 있다. 그러므로 Student와 Assistant는 특정한 관계가 있다고 판단할 수 있으며 이런 계층적 비교 방법을 이용하여 묵시적 참조 무결성 관계 정보를 획득 한다.

위에서 보여 진 결과들을 종합하여 각 컬럼 이름들 간의 관계를 살펴보면 표 5와 같다.

표 5에 나타난 후보 쌍들을 대상으로 VP-T에서 제안한 알고리즘을 이용하여 검증 단계를 수행한다. 외래키

**표 5.** 묵시적 참조 무결성 관계 후보 쌍

| |
|---|
| $RI_{can}$ = { (Student.StudentID, Professor.AssistantID),<br>(Room.OfficeID, Class.Place),<br>(Room.OfficeID, Professor.Office)} |

**표 6.** 출력 관계형 스키마 모델

| |
|---|
| **테이블 집합 T** |
| T = {Student, Professor, Class, Room} |
| **컬럼 집합 C(table_name)** |
| C(Student) = {StudentID, Sname, Pid, Cname} |
| C(Class) = {Cname, Place, Time} |
| C(Professor) = {Pid, Pname, Office, AssistantID} |
| C(Room) = {OfficeID, ManagerID, Rname} |
| **주키 컬럼 K(key_column)** |
| K(Student) = {StudentID} |
| K(Class) = {Cname} |
| K(Professor) = {Pid} |
| K(Room) = {OfficeID} |
| **컬럼들의 속성 P(column_name)** |
| P(StudentID) = {string, u, !n} |
| P(Sname) = {string, u, !n} |
| P(Pid) = {string, u, !n} |
| P(Cname) = {string, u, !n} |
| P(Cname) = {string, u, !n} |
| P(Place) = {string, u, !n} |
| P(Time) = {string, u, n} |
| P(Pid) = {string, u, !n} |
| P(Pname) = {string, u, !n} |
| P(Office) = {string, u, !n} |
| P(AssistantID) = {string, u, !n} |
| P(OfficeID) = {string, u, !n} |
| P(ManagerID) = {string, u, !n} |
| P(Rname) = {string, u, !n} |
| **$RI_{imp}$ = {(Student.StudentID, Professor.AssistantID),<br>(Room.OfficeID, Class.Place),<br>(Room.OfficeID, Professor.Office)}** |

제약 조건의 정의를 적용하여 비교 대상 컬럼의 값이 다른 컬럼에 여러 번 나타날 경우 이들은 묵시적 참조 무결성 관계를 가진다고 결정한다.

위에서 예시한 결과들을 이용하여 표 6에서 출력 관계형 스키마 모델을 추출한다.

## 5. 시스템 구조

그림 2는 묵시적 참조 무결성 관계를 추출하기 위한 프로세스 및 시스템의 각 구성 요소이다.

이 논문에서 제안한 시스템은 관계형 데이터베이스 스키마를 XML문서로 변환하기 위해 컬럼을 추출하고, 컬럼들의 유사도를 분석, 결과에 대한 타당성을 검사를 하는 프로세스를 가진다. 유사도 분석 엔진(SAE, Similarity Analysis Engine)은 비교(Comparison) 컴포넌트, 유사성 검사(Similarity) 컴포넌트, 클러스터링(Clustering) 컴포넌트, 그리고 검증(Verification) 컴포넌트로 구성된다. 각 구성요소들은 다음과 같은 역할을 가진다.

- 비교 컴포넌트
  - 특정 컬럼에 들어 있는 불필요 단어 제거
    - 'ID' : 직관적으로 알 수 있는 접두어 또는 접미사
  - 검색 하고자 하는 단어가 LLD에 저장되어 있는 것인지 확인
- 유사성 검사 컴포넌트
  - 워드넷을 이용한 컬럼 이름의 유사성 비교
    - 동의어 방법
    - 계층적 방법
- 클러스터링 컴포넌트
  - Local Lexical Dictionary에 유사 어휘 저장
    - 새로운 단어 검색 시 활용
- 검증 컴포넌트
  - VP-T알고리즘을 이용한 1:N 관계 검증

위의 4가지 컴포넌트를 이용하여 관계형 데이터베이스에서 XML 스키마로 변환 시 묵시적 참조 무결성 관계 정보를 추출할 수 있다.

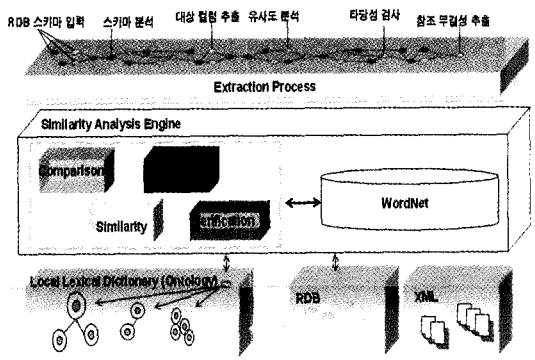



그림 2. 시스템 구성요소

## 6. 비교 평가

이 절에서는 논문에서 제안하고 있는 알고리즘에 대한 성능 평가를 위하여 관계형 데이터베이스의 XML 문서로의 변환을 위해 제안된 기존 알고리즘(CoT, VP-T)와의 비교 평가를 수행한다. 지금까지 관계형 데이터베이스의 정보를 XML문서로의 변환을 위해 많은 방법들이 있었지만, 참조 무결성 정보를 추출하는 관련 연구는 위의 2가지 방법에서만 가능하다. 이 절에서는 현재까지 제안된 기존 알고리즘들과 이 논문에서 제안하는 알고리즘간의 차이를 비교 분석하여 기술한다.

이 논문에서 제안하고 있는 알고리즘은 기존의 연구 방법인 VP-T에서 묵시적인 참조 무결성 관계를 추출 시 해결 할 수 없었던 문제를 해결하였다. 기존 VP-T 방법으로는 컬럼들의 이름이 다른 경우, 또는 비교 대상 컬럼의 값이 1:N이 아닌 1:0 또는 1:1의 관계를 가지고 있을 경우에 묵시적 참조 무결성 관계 정보를 반영하지 못하여, XML 문서로 변환 시 정보 손실이 발생한다. 제안 알고리즘을 적용할 경우 이와 같은 문제를 해결하여 변환 시 참조 무결성 손실률을 줄일 수 있다.

### 6.1 변환 XML 문서를 이용한 비교 평가

그림 1에 있는 관계형 데이터베이스의 정보를 XML 문서로 변환하면 다음과 같다.

(a) CoT를 이용한 변환

```
<!ELEMENT Student      (StudentID, Sname)>
<!ATTLIST Student Cname IDREF #REQUIRED>
<!ELEMENT Professor    (pname, Office, AssistantID,
Student)>
<!ATTLIST Professor pid ID #REQUIRED>
<!ELEMENT Class        (Place, Time)>
<!ATTLIST Class Cname ID>
<!ELEMENT Room     (OfficeID, Rname)>
<!!ATTLIST Room Ref_Pid IDREF>
```

(b) VP-T를 이용한 변환

```
<!ELEMENT Student      (StudentID, Sname, Pid)>
<!ATTLIST Student pid IDREF #REQUIRED>
<!ATTLIST Student Cname IDREF #REQUIRED>
<!ELEMENT Professor    (pname, Office, AssistantID)>
<!ATTLIST Professor pid ID #REQUIRED>
<!ELEMENT Class        (Place, Time)>
<!ATTLIST Class Cname ID #REQUIRED>
<!ELEMENT Room (OfficeID, Rname)>
<!ATTLIST Room Pid IDREF #REQUIRED>
```

(c) 제안 알고리즘을 이용한 변환

```
<!ELEMENT Student       (StudentID, Sname, Pid)>
<!ATTLIST Student pid IDREF #REQUIRED>
<!ATTLIST Student StudentID ID #REQUIRED>
<!ATTLIST Student Cname IDREF #REQUIRED>
<!ELEMENT Professor     (pname, Office, AssistantID)>
<!ATTLIST Professor pid ID #REQUIRED>
<!ATTLIST Professor AssistantID IDREF #REQUIRED>
<!ELEMENT Class         (Time)>
<!ATTLIST Class Place IDREF #REQUIRED>
<!ATTLIST Class Cname ID #REQUIRED>
<!ELEMENT Room (OfficeID, Rname)>
<!ATTLIST Room Pid IDREF #REQUIRED>
<!ATTLIST Room OfficeID ID #REQUIRED>
```

CoT 알고리즘은 명시적인 참조 무결성 관계 정보만을 반영한다. 따라서 CoT 알고리즘은 묵시적으로 정의된 참조 무결성 관계 정보는 보장할 수 없다. VP-T 알고리즘은 CoT알고리즘에서 반영된 명시적 참조 무결성 관계 정보뿐 만 아니라 묵시적 참조 무결성 관계 정보를 추출한다. 하지만 대상 컬럼의 이름이 다를 경우 컬럼들에 대한 비교를 할 수 없기 때문에 묵시적 참조 무결성 관계를 완벽하게 반영하지 못한다. 제안된 알고리즘은 같은 의미를 가지는 이질적인 단어에 대한 정보를 반영할 수 있기 때문에 변환 시 이전의 방법들 보다 정보 손실을 줄인다.

## 6.2 실험 데이터

이 논문에서는 MS Access Northwind Sample을 이용하여 기존 방법들과의 비교 분석을를 수행하였다. Northwind Schema는 8개의 테이블과 76개의 컬럼으로 구성되어 있다. 예제 데이터베이스의 참조 무결성은 모두 명시적으로 잘 정의 되어 있다. 따라서 보다 명확한 비교 평가를 위하여 실험 데이터 모델의 물리적인 관계성을 일부 제거하고, 동일한 의미를 가지는 컬럼의 이름을 다른 이름으로 수정하여 명시적 참조 무결성 관계를 일부 변경한다. 해당 샘플을 이용하여 변환 정확도와 관계 추출 시간에 대하여 실험하였다. 대상 컬럼의 이름이 비자연어이거나, 워드넷에 존재 하지 않는 경우, 동음이의어의 경우에는 유사성을 검사 할 수 없기 때문에 묵시적 참조 무결성 관계를 추출해 낼 수 없다. 실험에 이용한 샘플 데이터의 총 7개의 참조 무결성 관계 중 1개의 참조 무결성 관계는 컬럼 간 동일한 의미를 가지나 서로 다른 단어로 되어 있다. 다른 1개의 참조 무결성 관계에서 컬럼들의 값은 1:1의 관계를 가진다. 샘플 데이터를 요약하면 표 5와 같다.

**표 5.** 실험 데이터의 정보 및 제약 사항

| 항목 / 자료 | T | C | $RI_{all}$ | $RI_{no\text{-}value}$ | $RI_{exp}$ | $RI_{imp}$ | $RI_{equ}$ |
|---|---|---|---|---|---|---|---|
| North | 8 | 76 | 7 | 1 | 1 | 5 | 2 |

*T* : 테이블 수, *C* : 컬럼 수, RIall : 참조 무결성 관계 총 수
RIexp : 명시적 참조 무결성 관계 수
RIimp : 묵시적 참조 무결성 관계 수
RIno-value : 추출 불가능한 참조 무결성 관계 수
RIequ : 컬럼 명이 같지만 실 데이터 의미가 다른 경우

### 6.2.1 변환 정확도

XML 문서로의 변환 정확도는 변환 시 참조 무결성 관계 정보를 얼마나 정확히 반영하였는가를 보여준다.

그림 3은 데이터베이스의 변환 정확도를 나타낸다. CoT 는 변환 시 명시적 참조 무결성 관계만을 고려하므로 RIexp의 개수만큼 만 추출 가능하다. VP-T 알고리즘은 변환 시 묵시적인 참조 무결성 관계를 추출 할 수 있으나 1:N관계를 가지는 컬럼들만 추출이 가능하다. 위의 예제에서 VP-T는 제안하는 알고리즘 보다 참조 무결성 관계를 2개 더 추출하였다. 하지만 추가된 2개의 개수는 컬럼이름은 같으나 실제 데이터 자체로는 전혀 관계가 없는 것을 추출하였다. 그러므로 정확도를 살펴보면 VP-T는 55.6%의 정확도를, 제안하는 알고리즘은 71.4%의 정확도를 가진다. 즉, VP-T를 이용하여 제안하는 알고리즘보다 참조 무결성 관계를 많이 추출하였으나, 실질적인 정확도는 떨어짐을 확인 할 수 있다. 하지만 제안 알고리즘을 통해서 묵시적 참조 무결성 관계 정보를 모두 추출하지 못할 수 있다. 그 결과 위에서 6개가 아닌 5개만 을 추출하였다. 결과가 이와 같이 나온 이유는 많은 용어들에 대한 온톨로지가 구축 되어 있는 워드넷에서 구축되지 않은 어휘들을 검색 할 때 유사성 검사를 수행하지 못했기

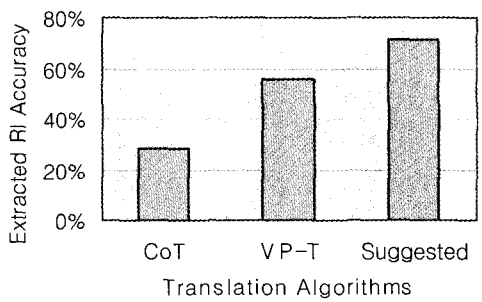



그림 3. 변환 정확도

때문이다. 또한 실제로 논리데이터에 묵시적 참조 무결성 관계를 가지고 있지만 컬럼의 유사성 검사에서 연관 관계가 거의 없다고 나올 경우 VP-T알고리즘 보다 손실률이 클 수 있다. 하지만 이런 경우 유사성 검사 기준을 반대로 적용하여 유사도가 거의 없는 컬럼 명을 대상에서 제거한 후 관계성이 큰 컬럼들을 계층적 방법을 이용하면 위의 문제점을 보완할 수 있다.

#### 6.2.2 관계 추출 시간에 따른 비교

이 논문에서 제안 하는 알고리즘을 이용하는 경우 VP-T 방법 보다 빠른 시간 안에 관계 정보를 추출할 수 있다. VP-T는 비교 대상의 컬럼을 추출하여 컬럼 간의 값을 비교해야 한다. 그렇기 때문에 대용량 데이터베이스를 쓰고 있는 경우 컬럼이 가지고 있는 값의 개수만큼 비교하는 시간이 비례하여 기하급수적으로 증가한다. 그러나 이 논문에서 제안한 알고리즘은 워드넷을 이용한 컬럼 유사성 비교를 통해 비교 대상이 되는 컬럼을 선추출한다. 따라서 제안 알고리즘은 기존의 알고리즘들에 비해 컬럼 간 비교 횟수를 줄일 수 있으며 관계형 데이터의 XML 데이터 변환 시 보다 빠른 속도로 관계 정보를 추출해 낼 수 있다.

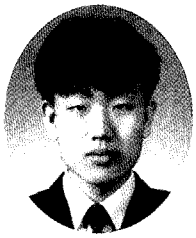
## 7. 결론 및 향후 연구

이 논문에서는 관계형 데이터베이스를 XML 문서로 좀 더 정확하고 효과적인 변환을 위한 알고리즘을 제안한다. 다른 이름으로 표현된 컬럼 간의 묵시적 참조 무결성 관계를 추출하기 위해 워드넷을 이용하여 컬럼들의 이름에 대한 유사성을 비교 한다. 이 논문의 결과를 통하여 설계자들은 묵시적 참조 무결성을 가질 수 있는 후보 컬럼들에 대한 정보를 획득함으로써 XML문서로 변환 시 관계형 데이터베이스에서 가지고 있는 의미정보의 손실을 방지 할 수 있다. 장점을 요약해 보면 다음과 같다.

이 논문에서는 두 컬럼의 이름에 대한 유사성 비교를 위하여 해당 컬럼의 이름은 약어를 사용하지 않고 전체 이름으로 구성된 한 개의 단어라고 가정하였다. 향후 연구에서는 컬럼들의 이름이 축약되어 있어 워드넷으로부터 동의어를 추출하지 못하는 경우나, 동음이의어와 같은 경우에 대해서는 향후 연구 과제로 남겨둔다.

## 참 고 문 헌

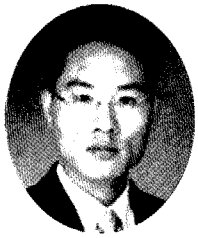

1. T. Bray, J. Paoli, and M. Cavary, Extensible Markup Language (XML) 1.0, 2nd Edition, W3C Recommendation, October, 2000.
2. R. Elmasri, and S. Navathe, Fundamental of Database Systems, 4th Edition, Addison-Wesley, 2003.
3. D. Lee, M. Mani, F. Chiu, and W. W. Chu, "NeT&CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints", 11th ACM Int'l Conference on Information and Knowledge Management (CIKM). McLean, VA, USA, November, 2002.
4. Jinhyung Kim, Dongwon Jeong, and Doo-Kwon Baik, "An Algorithm for Automatic Inference of Referential Integrities during Translation from Relational Database to XML Schema", Springer Verlag, Lecture Notes in Artificial Intelligence (LNAI), Vol. LNAI 3802, pp.165-170, December 2005.
5. G. A. Miller, "WordNet : An On-Line Lexical Database", International Journal of Lexicography, 1990s.
6. 이재윤, 김태수 (1998) "워드넷과 시소러스" 언어정보개발연구 창간호, 225-228.
7. D. Lee, M. Mani, F.Chit, and W. W. Chu, "Effective Schema Conversions between XML and Relational Models", European Conference on Artificial Intelligence (ECAI).
8. D. Lee, M. Mani, F.Chit, and W. W. Chu, "Nesting-based Relational-to-XML Schema Translation", Int'l Workshop on the Web and Databases (WebDB), Santa Barbara, CA, May 2001.



**김 장 원** (jwkim@software.korea.ac.kr)

1998년 상명대학교 컴퓨터소프트웨어 학사
2006년~현재 고려대학교 컴퓨터 학과 석사

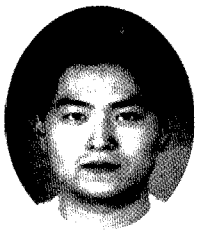
관심분야 : 데이터베이스, XML, 모델링&시뮬레이션



**정 동 원** (djeong@kunsan.ac.kr)

1997년 군산대학교 컴퓨터과학과(이학사)
1999년 충북대학교 전산과(이학석사)
2004년 고려대학교 컴퓨터학과(이학박사)
1999년~2000년 ICU 부설 한국정보통신교육원(전임강사)
2000년~2001년 지구넷 부설 연구소(선임연구원)
2002년~2005년 라임미디어 테크놀로지(연구원)
2004년 고려대학교 정보통신기술연구소(연구교수)
2005년 펜실베니아 주립대학(PostDoc.) 2002년~현재 TTA 표준화위원회-메타데이터표준화 프로젝 그룹 PG406(특별위원)
2005년~현재 군산대학교 정보통계학과(교수)
2006년~현재 ISO/IEC JTC1/SC32 국내위원회(위원)

관심분야 : 데이터베이스, 이동 에이전트 시스템 및 보안, 유비쿼터스 컴퓨팅



**김 진 형** (koolmania@software.korea.ac.kr)

2004년 홍익대학교 컴퓨터공학과(공학사).
2006년 고려대학교 컴퓨터학과(이학석사).
2006년~현재 고려대학교 컴퓨터학과 박사과정

관심분야 : 데이터베이스, XML, 유비쿼터스 컴퓨팅

**백 두 권** (baik@software.korea.ac.kr)

1974년 고려대학교 수학과(이학사)
1977년 고려대학교 대학원 산업공학과(공학석사)
1983년 Wayne State Univ.(전산학석사)
1985년 Wayne State Univ.(전산학 박사)
1986년~현재 고려대학교 컴퓨터학과(교수)
1989년~현재 한국정보과학회(이사/평의원/부회장)
1991년~현재 한국시뮬레이션학회(이사/부회장/감사)
1991년~현재 ISO/IEC JTC1/SC32 국내위원회(위원장)
2002년~2004년 고려대학교 정보통신대학(초대학장)
2002년~2004년 한국시뮬레이션학회(회장)
2001년~현재 행자부 등록 (사)도산아카데미(원장)
2004년~현재 정통부 등록 (사)한국정보처리학회(부회장)
2005년~현재 정통부 등록 (사)한국정보과학회(부회장)
2005년~현재 교육부 등록 (사)홍사단(공의원)
2005년~현재 산자부 등록 (사)한국시뮬레이션학회(고문).

관심분야 : 데이터베이스, 소프트웨어 공학, 시뮬레이션, 메타데이터, 정보 통합, 정보 보호 등