시맨틱 검색을 위한 이기종 데이터간의 매칭방법

Matching Method between Heterogeneous Data for Semantic Search

이기정, 황보택근
경원대학교 전자계산학과

Ki-Jung Lee(jcm5758@empas.com), Taeg-Keun Whangbo(tkwhangbo@kyungwon.ac.kr)

요약

시맨틱 환경에서의 시맨틱 검색을 위해서는 분산된 자원의 관리와 처리가 중요한 요소이다. 분산된 자원의 효율적인 검색을 위해서는 온톨로지의 사용이 필수적이지만, 모든 자원에 대한 통합적인 온톨로지를 구축하는 것은 현실적으로 매우 어려운 일이다.

본 논문에서는 웹 환경에서의 대부분의 자원은 관계형 데이터베이스 형태로 저장되어져 있다고 가정하고, 시맨틱 검색을 위하여 분산된 관계형 데이터베이스 테이블과 도메인 온톨로지간의 매칭을 위한 방법을 제안한다. 기존의 관계형 데이터베이스와 도메인 온톨로지간의 매칭에 관한 연구들은 관계형 데이터베이스에서 로컬 온톨로지를 추출하여 도메인 온톨로지와의 매칭을 수행하였다. 그러나, 로컬 온톨로지를 추출하는 과정에서 도메인 온톨로지와의 상관관계를 이용하지 않음으로 인하여 도메인 정보가 손실되는 문제점을 가지고 있다. 이에 대한 해결책으로 관계형 데이터베이스의 인스턴스들과 도메인 온톨로지의 인스턴스간의 유사도 측정을 통한 정보 손실을 방지하였으며, 관계형 데이터베이스내의 테이블들간의 관계와 온톨로지에서의 클래스들간의 관계 정보를 이용하여 보다 효율적인 매칭이 가능하도록 하였다.

■ 중심어 : | 온톨로지 매칭|스키마 매칭|인스턴스 매칭|이기종 데이터 매칭|

Abstract

For semantic retrieval in semantic web environment, it is an important factor to manage and manipulate distributed resources. Ontology is essential for efficient search in distributed resources, but it is almost impossible to construct an unified ontology for all distributed resources in the web.

In this paper, we assumed that most information in the web environment exist in the form of RDBMS, and propose a matching method between domain ontology and the existing RDBMS tables for semantic retrieval. Most previous studies about matching between RDBMS tables and domain ontology have extracted a local ontology from RDBMS tables at first, and conducted the matching between the local ontology and domain ontology. However in the processing of extracting a local ontology, some problems such as losing domain information can be occurred since its correlation with domain ontology has not been considered at all. In this paper, we propose a methods to prevent the loss of domain information through the similarity measure between instances of RDBMS tables and instances of ontology. And using the relational information between RDBMS tables and the relational information between classes in domain ontology, more efficient instance-based matching becomes possible.

■ keyword : | Ontology Matching|Scheme Matching|Instance Matching|Heterogeneous Matching|

* 본 연구는 제2단계 BK21사업에 의하여 지원 받았습니다.
접수번호 : #060809-001
접수일자 : 2006년 08월 09일
심사완료일 : 2006년 09월 15일
교신저자 : 이기정, e-mail : jcm5758@empas.com

I. 서 론

기존의 검색 시스템은 두 가지의 문제점을 가지고 있다. 첫째, 사용자의 질의를 분석하여 검색을 수행할 수 없다. 기존의 검색 시스템은 사용자가 질의한 키워드를 위주로 키워드가 포함된 검색 결과만을 사용자에게 반환하였다. 또한, 검색 키워드간의 관계를 해석할 수 없었기 때문에 검색 키워드가 모두 혹은 둘 중의 하나가 포함된 결과만을 제시한다. 이러한 문제점을 해결하기 위하여 시맨틱 웹을 이용한 의미 검색이 필요하며, 이에 대한 연구가 활발히 진행중이다[1][2]. 그러나, 시맨틱 웹을 이용하기 위해서는 해당 도메인에서 사용하고 있는 중요한 개념이나 어휘 등이 온톨로지 형태로 구축이 되어져야만 의미 검색이 가능하며, 이 온톨로지를 각 기관별로 구축하기 위해서는 전문가의 도움이 필요하다.

둘째, 문화재와 같이 특수한 경우에는 각 기관별로 별도로 데이터베이스를 구축하며, 이를 통합하여 검색할 수 있는 검색 시스템이 아직까지 존재하지 않는다. 이를 위해서는 각 기관별로 구축된 데이터베이스를 분석하여 통합적인 검색이 가능하도록 데이터베이스 구조를 통일하거나 검색을 위한 중앙 검색 시스템을 만들어서 각 기관별 데이터베이스와 매칭하는 방법을 사용할 수 있다. 통일된 구조를 만드는 것은 많은 비용과 시간이 소요되기 때문에 현실적으로 불가능하며, 중앙 검색 시스템과 매칭하는 방법 역시 많은 비용이 소요된다.

본 논문은 분산된 데이터베이스를 통합하여 검색하고 의미 검색이 가능하도록 하기위하여 로컬 데이터베이스와 도메인 온톨로지간의 매칭을 목표로 한다. 이를 위하여 두 가지 전제 조건을 이용한다. 첫째, 각 기관별 데이터베이스는 관계형 데이터베이스를 이용한다라는 것이고, 둘째 중앙 검색 시스템에는 문화재와 관련된 도메인 온톨로지를 가지고 있다는 것이다.

두 가지 전제 조건을 바탕으로 본 논문에서는 통합 검색을 위한 2단계 방법을 사용한다. 첫 번째 단계에서는 각 기관별로 구축된 데이터베이스에서 로컬 온톨로지 를 추출한다. 각 기관별 데이터베이스내의 테이블 정보들과 도메인 온톨로지를 활용하여 각 기관별로 로컬 온톨로지를 추출한다. 기존 연구들은 데이터베이스 내의 정보만을 사용함으로 데이터베이스내의 정보를 추출하는 데 한계를 가지고 있다. 따라서, 본 논문에서는 도메인 온톨로지를 활용하여 이러한 단점을 극복한다. 두 번째 단계에서는 추출된 각 기관별 로컬 온톨로지와 도메인 온톨로지간의 매칭을 통하여 통합 검색을 위한 기반 요소를 구축한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 관계형 데이터베이스에서 메타정보 추출 방법과 클래스 후보군 추출 방법에 대하여 논의하고, 4장에서는 관계형 데이터베이스와 도메인 온톨로지간의 상세 매칭 방법에 대하여 살펴본다. 5장에서는 실험을 통해 제안한 알고리즘의 성능을 평가하고, 마지막으로 6장에서는 결론 및 향후 연구 과제에 대하여 언급한다.

II. 관련 연구

2.1 시맨틱 웹과 온톨로지

시맨틱 웹은 웹 상의 정보에 컴퓨터가 쉽게 해석할 수 있는 잘 정의된 의미를 부여하여 사람과 컴퓨터간의 협동작업을 원활하게 하기 위해서 제안되었다[1-2]. 웹 상에 존재하는 정보를 사람뿐만 아니라 컴퓨터도 해석할 수 있도록 하는 것에 그 목적이 있다. 앞장에서 예를 든 것처럼 사용자의 질의를 분석하여 사용자가 원하는 정보를 추출하기 위해서는 시맨틱 웹을 이용한 검색 즉 의미 검색 기술이 필요하다.

시맨틱 웹은 3가지 요소로서 구성된다. 첫째, 웹 상에 존재하는 자원을 기술하는 자원 서술 즉 메타 정보, 둘째 자원에 대한 의미 부여 혹은 자원과 자원의 관계를 표현하기 위한 온톨로지, 마지막으로 온톨로지를 이용하여 사용자가 요구하는 결과를 만들기 위한 추론으로 이루어진다[3].

온톨로지는 메타 정보와 메타 정보간의 관계를 나타낸 것으로, 클래스, 데이터 프로퍼티(data property), 오브젝트 프로퍼티(object property) 등의 요소를 가지

고 있다. 클래스는 [그림 2]에 나타낸 것처럼 사찰, 탑, 용도 등의 자원을 개념화한 것이다. 데이터 프로퍼티는 자원의 속성을 의미하는 것으로 사찰명, 창건자 등이 있다. 오브젝트 프로퍼티는 클래스들간의 관계를 나타내는 것으로 도메인(domain)과 범위(range)를 가진다. 오브젝트 프로퍼티의 도메인은 관계의 주체이며, 도메인 온톨로지내의 클래스를 나타낸다. 범위는 관계의 적용 범위를 나타낸다. 예를 들어, "hasStupa(Temple, Stupa)"라는 오브젝트 프로퍼티가 존재한다면 이는 Temple 클래스를 도메인으로하고, Stupa 클래스를 범위로 갖음으로써 Temple 클래스가 Stupa 클래스를 포함한다는 의미가 된다.

본 논문에서는 의미 검색을 위하여 각 기관별로 존재하는 데이터베이스에서 자원을 추출하고, 각 자원과 자원간의 관계를 설정하여 로컬 온톨로지를 추출하고, 추출된 로컬 온톨로지와 도메인 온톨로지간의 매칭을 수행하는 방법을 사용한다.

2.2 온톨로지 추출과 매칭

관계형 데이터베이스와 온톨로지간의 매칭에 관한 연구는 주로 데이터베이스에서 온톨로지를 추출하여 도메인 온톨로지와의 매칭을 수행하는 방법을 사용하였다.

관계형 데이터베이스에서 온톨로지를 추출하는 연구는 테이블 명칭을 이용하는 방법과 ER 데이터모델을 이용하는 방법으로 구분된다. 테이블 명칭을 이용하는 방법[4]은 테이블의 명칭과 온톨로지 클래스 레이블들을 비교하여 유사한 것들끼리 매칭하는 알고리즘을 사용한다. 하지만 이 경우 명칭이 유사하지 않거나, 같은 명칭이라도 다른 의미의 데이터일 경우 매칭의 정확도가 저하된다.

An[5]과 Gramajo[6]는 ER 데이터모델을 이용하여 엔티티의 종류를 strong과 week로 분류했으며, 이들의 관계를 함수로 표현하여 온톨로지를 구축하는 방법을 사용하였다. 그러나, 관계형 데이터베이스를 구축하였다고 하더라도 ER 데이터모델을 사용하지 않았거나 현재 가지고 있지 않은 경우에는 이를 적용하기 어려운 단점이 있다. Upadhyaya[7]는 확장형 ER 데이터모델과 필드의 속성 정보를 이용하여 온톨로지 추출을 수행하였다. Li[8]는 ER 데이터모델에서 주로 키 정보 즉 주키(primary key)와 외래키(foreign key) 정보들을 이용하여 테이블들간의 관계를 만드는 연구를 수행하였다. Trinh[9]는 ER 데이터모델을 사용하지 않고 데이터베이스에서 메타데이터와 제약조건들을 추출하였으며, 추출된 정보들을 별도로 정의한 용어들과의 관계를 통해서 온톨로지를 생성하였다.

이러한 연구들은 관계형 데이터베이스에서 온톨로지 요소를 추출함에 있어서 도메인 온톨로지의 정보를 활용하지 않고 각 데이터베이스들의 정보만을 활용하였다. 따라서, 각 데이터베이스에서 온톨로지를 추출하였다는 의미는 갖겠지만 도메인 온톨로지와의 연관성을 상실하게 된다.

본 논문에서는 이러한 단점을 보완하기 위해서 각 기관별 로컬 온톨로지 추출시 도메인 온톨로지와의 연관성을 활용하였으며, 이를 이용하면 로컬 온톨로지와 도메인 온톨로지간의 매칭의 정확성도 향상된다는 것을 실험을 통해서 나타내었다.

2.3 용어 정의

본 논문에서는 관계형 데이터베이스에서 추출된 클래스를 C_D(클래스명)으로 표현하고, 도메인 온톨로지에서 사용된 클래스를 C_O(클래스명)로 표현한다. 예를 들어, 관계형 데이터베이스에 추출된 클래스명이 Stupa라면 C_D(Stupa)로 표현한다. 데이터 프로퍼티의 경우 DP_D(클래스명, 속성명)과 DP_O(클래스명, 속성명)으로 표현한다. 오브젝트 프로퍼티는 OP_D(도메인명, 범위) 그리고 OP_O(도메인명, 범위)로 표현한다.

2.4 사용한 데이터

본 논문에서 사용한 ER 데이터모델을 [그림 1]에 나타내었다. [그림 1]의 데이터모델은 사찰과 탑에 대한 단순한 형태의 데이터모델을 설계한 것으로, 사찰(Temple), 탑(Stupa), 용도(ForUse), 지정(DegiType), 문양(PatType) 등의 5개의 테이블을 사용하였다. Temple과 Stupa 테이블은 사찰과 탑에 대한 기본 정보들을 담고 있으며, ForUse, DegiType,

PatType 테이블들은 Temple과 Stupa 테이블이 가질 수 있는 부분 정보들을 담고 있다. 또한, Temple과 Stupa 테이블내에는 시대(CreatePeriod), 소유구분(OwnType) 등과 같이 별도로 독립 가능한 형태의 데이터들도 존재하도록 설계되었다. 이는 복잡한 형태의 관계형 데이터베이스에 대한 실험을 위하여 사용되었다.

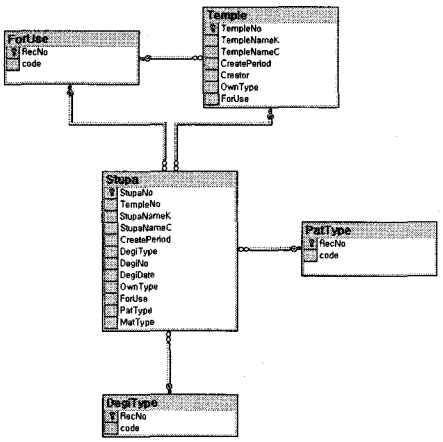


그림 1. 사용한 ER 데이터모델

[그림 2]는 본 논문에서 사용한 온톨로지의 일부를 나타내고 있다. 온톨로지는 이해를 돕기 위하여 한글로 작성되었으며, 실제 온톨로지 클래스 명칭은 영어로 작성되었고, 한글 명칭은 레이블을 이용하여 표현하였다. 도메인 온톨로지는 사찰과 탑에 관련된 정보를 담고 있으며, 총 7개의 최상위 클래스들로 구성되어 있다. 각 클래스들은 하위 클래스들과 데이터 프로퍼티, 오브젝트 프로퍼티를 가지고 있다. 각 클래스들의 인스턴스들은 국립박물관[10], 문화재청[11], 그리고 한국의 전통 사찰[12] 홈페이지에서 추출하여 사용하였다.

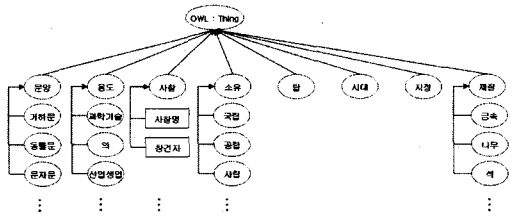


그림 2. 사용한 온톨로지

III. 관계형 데이터베이스에서의 온톨로지 추출

제안한 시스템의 흐름도는 [그림 3]과 같다. 관계형 데이터베이스에서 로컬 온톨로지를 추출하고, 로컬 온톨로지와 도메인 온토로지간의 매칭을 수행하여 매칭 정보를 추출한다.

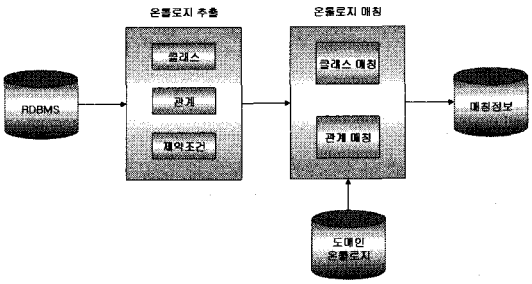


그림 3. 시스템 흐름도

관계형 데이터베이스 테이블과 온톨로지와의 매칭을 위해서 먼저, 데이터베이스 메타 정보를 추출이 필요하다. 이 과정에서는 ER 데이터모델을 기반으로 클래스 후보군과 데이터 프로퍼티 후보군, 그리고 테이블들간의 관계를 추출한다.

클래스 추출과정은 2단계로 이루어진다. 첫 번째 단계에서는 관계형 데이터베이스의 테이블 메타정보를 이용하여 추출하는 것이고, 두 번째 단계에서는 테이블 내의 필드 정보를 이용하여 추출하는 것이다.

기존 연구들의 경우 테이블 정보만 이용하기 때문에 테이블내에 존재하는 하위 클래스 개념은 추출하지 못하는 단점을 가지고 있다. 이를 보완하기 위하여 테이블들의 필드 정보들을 추출하여 도메인 온톨로지와의 클래스 매칭을 수행하며, 클래스 매칭의 결과로 테이블 정보에서 추출할 수 없는 하위 클래스 개념들을 추출할 수 있다.

1. 메타 정보를 이용한 클래스 추출과정

관계형 데이터베이스의 메타정보에는 테이블 구조와 필드 정보, 키 정보 등이 담겨져 있으며, 이의 분석을 통해 클래스 후보군을 추출한다. 먼저, 각 테이블들을

클래스 후보군으로 선정한다. 본 논문에서는 데이터베이스내의 모든 테이블들을 클래스 후보군으로 선정한다. 그 이유는 일반적으로 데이터베이스내의 모든 테이블은 각각의 필요성에 의하여 작성되었을 것으로 추정하기 때문이다. C_D(Temple), C_D(Stupa), C_D(ForUse), C_D(PatType), C_D(DegiType)의 5개의 클래스가 생성된다.

2. 유사도 비교를 통한 클래스 후보군 추출 과정

테이블내의 필드들은 클래스와 데이터 프로퍼티로 분류될 수 있다. 필드들은 해당 테이블의 상세 속성을 표현할 수 있고, 혹은 코드 정보와 같은 독립 가능한 클래스로 표현될 수 있다. 예들 들어, [그림 1]에서 Stupa 테이블의 StupaNameK 필드는 탑의 명칭을 나타내는 속성으로 데이터 프로퍼티로 간주될 수 있다. 하지만, 데이터베이스의 메타 정보만으로는 이러한 특성을 구분할 수 없기 때문에 본 논문에서는 필드의 인스턴스를 통해서 이를 구별하였다. 이를 위해 각 테이블내의 필드 인스턴스들과 도메인 온톨로지내의 말단 클래스들의 인스턴스간의 유사도 비교를 통해서 클래스 후보군을 추출한다. 유사도 비교 값이 일정 값 이상이면 이를 클래스 후보로 선정하고 그렇지 않은 경우 이를 데이터 프로퍼티로 분류한다.

또한, 유사도 비교를 수행하면 테이블내의 독립 가능한 필드들을 클래스로 추출할 수 있다. 예를 들어, Stupa 테이블의 경우 CreatePeriod를 가지고 있는데, 이 필드의 경우 시대 정보를 저장하고 있다. 일반적으로 시대 정보들은 코드 형식의 특정 데이터로 구분이 가능하다. 따라서, 도메인 온톨로지 내의 시대 클래스와 인스턴스 비교를 수행하면 C_D(CreatePeriod)를 추출할 수 있다.

유사도 비교 과정에서는 기본적으로 모든 테이블의 모든 필드에 대해서 수행하는 것을 기본으로 하지만 필드가 주키로 지정되어 있고, 다른 테이블의 참조 필드이면서 일련의 숫자인 경우에는 유사도 비교를 생략한다. 외래키의 경우 다른 테이블의 데이터를 참조로 사용하고, 주키의 경우 일반적으로 인덱싱의 용도로 사용되기 때문이다.

유사도 측정은 Levenshtein Distance(LD)[13]와 Longest Common Prefix/Suffix(LCPS)[13]를 이용하였다. LD는 문자 s에서 t로의 변환을 위해서 삽입, 삭제, 대체 등에 필요한 회수를 계산하는 방법으로 Edit Distance로 표현되기도 하며, 식(1)과 같이 표현된다.

$$LD(s,t) = 1 - \frac{changeLength}{\max Length} \quad (1)$$

식(1)에서 maxLength는 max(length(s), length(t))로 표현되며, changeLength는 변환을 위해 필요한 회수를 의미한다. 즉, 얼마나 많이 변경하여 두 문자가 같아질 수 있는 지를 측정하는 함수이다. 예를 들어, "통일신라시대"와 "통일신라"는 2번의 삽입을 통하여 동일한 문자가 된다.

LCPS는 접두사 혹은 접미사의 일치하는 숫자를 이용하여 비교하는 방법이다. 식(2)는 접두사를 통한 유사도이며, 식(3)은 접미사를 통한 유사도이다. 본 논문에서는 식(4)와 같이 접두사와 접미사 중 값이 큰 것을 이용하였다.

$$LCP(s,t) = \frac{prefixLength}{\min Length} \quad (2)$$

$$LCS(s,t) = \frac{suffixLength}{\min Length} \quad (3)$$

$$LCPS(s,t) = \max(LCP(s,t), LCS(s,t)) \quad (4)$$

식(2)에서 prefixLength는 일치하는 접두어의 길이이고, minLength는 min(length(s), length(t))이다. 식(3)에서 suffixLength는 일치하는 접미어의 길이를 나타낸다.

유사도 측정은 데이터베이스 테이블의 모든 필드와 도메인 온톨로지의 모든 말단 노드의 인스턴스와 수행하며 식(5)를 이용한다. LD와 LCPS는 사용자에 의하여 비중이 다르게 적용된다. 최종적으로 식(6)에 의하여 유사도 측정값이 산출된다.

$$SD(s,t) = \alpha \times LD(s,t) + (1-\alpha) \times LCPS(s,t) \quad 0 \le \alpha \le 1 \quad (5)$$

$$GSD_k = \frac{1}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} SD(s_i, t_j) \quad (6)$$

$SD(s_i, t_j)$는 각 매칭쌍에 대한 유사도이며, 이는 LD와 LCPS에 별도의 가중치를 적용하여 계산한다. 유사도가 일정 정도를 넘는 요소들은 클래스 후보로 분류되고, 그렇지 않은 요소들은 데이터 필드로 분류된다. [표 1]에 유사도 측정에 의한 클래스 매칭 결과를 나타내었다.

표 1. 클래스 매칭 결과

테이블명	필드명	매칭 클래스
DegiType	code	지정
ForUse	code	산업생업
PatType	code	동물문
Stupa	CreatePeriod	시대
Stupa	OwnType	국립
Stupa	MatType	나무
Temple	CreatePeriod	시대
Temple	OwnType	국립

3. 키 정보를 이용한 오브젝트 프로퍼티 추출

일반적으로 관계형 데이터베이스에서는 외래키(foreign key)를 사용하여 다른 테이블과의 관계를 나타낸다. 이 관계는 온톨로지의 Is-A 관계이거나 혹은 오브젝트 프로퍼티 관계일 수 있다. Is-A 관계는 그림 2에서 재질 클래스와 석 클래스의 관계처럼 상속 개념을 나타내고, 오브젝트 프로퍼티는 "사찰 클래스가 탑을 가지고 있다"라는 소유의 개념을 나타낸다.

본 논문에서는 외래키 정보를 이용하여 오브젝트 프로퍼티 관계를 설정하였다. 외래키는 외부의 다른 테이블 정보를 참조하는 것으로 [그림 1]의 Stupa 테이블의 ForUse 필드에서 ForUse 테이블의 code 필드를 참조하였다. 그러나, 외래키를 한 방향으로의 관계만으로 해석할 수 없다. 예를 들어, [그림 1]에서 ForUse 테이블과 Stupa 테이블은 외래키로 정의되어 있는데, 이 경우에 Stupa 테이블이 ForUse 테이블을 가지는 경우로 표현할 수 있다. 그러나, Temple 테이블과 Stupa 테이블의 경우 Stupa가 Temple을 소유하는 것이 아니라 Temple이 Stupa를 소유하는 것이 가능하다. 따라서, 이 단계에서는 양방향으로 관계가 성립되도록 후보 관계를 설정하고, 도메인 온톨로지와의 관계 유사도 비교시에 이에 대한 보정을 수행한다.

[그림 1]에 대한 오브젝트 프로퍼티 생성 결과의 일부를 [표 2]에 나타내었다.

표 2. 생성된 오브젝트 프로퍼티

Relation	Domain	Range
hasClass	Stupa	ForUse
	Stupa	PatType
	Stupa	DegiType
	Stupa	Temple
	Stupa	CreaetPeriod
	Stupa	OwnType
	Stupa	MatType
	Temple	Stupa
	Temple	CreatePeriod
	Temple	OwnType
	Temple	ForUse

Ⅳ. 온톨로지 매칭

지금까지 추출한 클래스 후보군, 데이터 프로퍼티, 그리고 오브젝트 프로퍼티를 이용하여 도메인 온톨로지와의 유사도 비교를 수행하고, 이를 통해 클래스 후보군에서 클래스를 선정한다. 또한, 오브젝트 프로퍼티 매칭을 통해서 클래스들간의 관계를 비교하여 최종적인 매칭 정보를 생성한다.

1. 클래스 선정

이전 장에서 수행한 클래스 유사도 비교는 데이터베이스 필드 인스턴스와 도메인 온톨로지의 말단 노드의 인스턴스간의 비교를 통해서 클래스 후보군을 추출하였다. 클래스 후보군은 단순히 말단 노드와의 유사도 비교만을 수행하였기 때문에 상위 노드 즉 상위 클래

스와의 유사도 비교가 필요하다.

예를 들어, Stupa 테이블의 MatType 필드가 [그림 4]의 석 클래스와 매칭이 되었다고 가정하면, MatType 필드는 재질 클래스와의 매칭 가능성을 내포하게 된다. 따라서, 이 경우 MatType 필드의 인스턴스와 재질 클래스의 모든 하위 클래스들(금속, 나무, 석)의 인스턴스들과의 유사도 비교를 수행한다. MatType 필드의 인스턴스가 90%이상 재질 클래스의 인스턴스와 매칭되면 MatType 필드는 재질 클래스와 매칭되도록 변경되며 그렇지 않은 경우에는 석 클래스와의 매칭으로 유지된다.

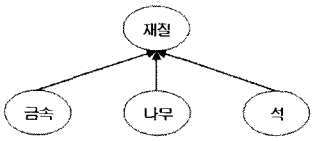


그림 4. 도메인 온톨로지의 재질 클래스

이 과정을 [그림 5]에 나타내었으며, 유사도 측정은 3장에서의 유사도 측정과 같은 방법을 사용하였으며, 갱신된 클래스 매칭 결과를 [표 3]에 나타내었다.

```
FOR each class of DB class
   Set co to matched class of class
   IF co has sibling class and co is not root class THEN
      IF InstanceMatching(class, parent of co) THEN
         Update matched class to parent of co
      END IF
   END IF
End FOR
```

그림 5. 부모 클래스 비교 프로그램

표 3. 갱신된 클래스 매칭 결과

테이블명	필드명	매칭 클래스
DegiType		지정
ForUse		용도
PatType		문양
Stupa	CreatePeriod	시대
Stupa	OwnType	소유
Stupa	MatType	재질
Temple	CreatePeriod	시대
Temple	OwnType	소유

2. 클래스와 오브젝트 프로퍼티를 이용한 매칭

관계형 데이터베이스와 도메인 온톨로지의 매칭에서 마지막 과정은 클래스와 오브젝트 프로퍼티를 이용하여 매칭하는 과정이다.

현재까지, Stupa 테이블과 Temple 테이블은 클래스 매칭이 되지 못하였다. 그 이유는 Stupa 테이블과 Temple 테이블의 일부 필드들은 클래스로 분리되었으며, 테이블내의 인스턴스를 비교할 수 없었기 때문이다.

즉, 테이블내에 많은 필드를 가지고 있는 경우 이 테이블들은 매칭이 수행되지 않는 경우가 발생한다. 이를 위해서 지금까지 매칭된 클래스 정보와 오브젝트 프로퍼티 정보를 이용하여 매칭되지 않는 클래스들을 추출한다.

Stupa 테이블의 경우 시대, 문양, 재질 클래스와 오브젝트 프로퍼티 관계를 가지고 있다. 도메인 온톨로지의 탑 클래스는 시대, 소유, 재질, 문양 등의 클래스들과 오브젝트 프로퍼티 관계를 가지고 있다. 이를 [그림 6]에 나타내었다.

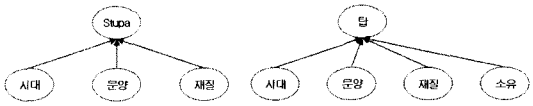


그림 6. Stupa 클래스와 탑 클래스

이를 매칭하기 위해서 도메인 온톨로지의 각 클래스들을 탐색하여 Stupa 클래스와 유사한 오브젝트 프로퍼티를 가지는 클래스를 찾아서 매칭 정보를 지정한다.

[그림 7]에 오브젝트 프로퍼티 비교에 대한 의사코드를 나타내었다.

```
FOR each class of DB classes
   Set cdop to object property of DB class
   FOR each class of Ontology classes
      Set coop to object property of ontology class
      IF compare(cdop, coop) THEN
         CALL matching(cdop, coop)
      END IF
   END FOR
End FOR
```

그림 7. 의사코드 : 오브젝트 프로퍼티 비교

V. 실 험

제안한 알고리즘의 효율성을 평가하기 위하여 3개의 관계형 데이터베이스를 생성하였다. 관계형 데이터베이스는 [그림 1]에서 제시한 데이터베이스와 문화재청[11]과 전통사찰관광종합정보[12] 홈페이지에서 제공하고 있는 문화재 검색 서비스의 자료를 이용하였다.

매칭 성능을 비교하기 위하여 [그림 2]에서 제시한 도메인 온톨로지를 이용하여 각 데이터베이스와 매칭 정확도를 계산하였다.

실험 방법은 각각의 관계형 데이터베이스에서 추출된 클래스들과 도메인 온톨로지간의 일대일 매칭을 수행하여 매칭 결과를 계산하였다. 정확도 계산은 미리 정의한 매칭 정보를 이용하여 추출된 클래스의 매칭 정보가 일치하는 지를 비교하였다. 즉, (매칭 정확도 = 올바른 매칭 / 추출된 클래스 수)로 계산하였다.

실험의 결과를 [표 4]에 나타내었으며, 평균적으로 84%의 정확도를 나타내었다. [표 4]에서 테이블 수는 관계형 데이터베이스에서 사용한 테이블의 수이며, 추출된 클래스 수는 알고리즘 적용 후 클래스로 추출된 수이다.

표 4. 실험 결과

	테이블 수	추출된 클래스 수	매칭 정확도(%)
DB1	5	9	88.8%
DB2	6	12	83.3%
DB3	10	21	80.9%
평 균			84.3%

잘못된 매칭의 경우 클래스 추출 시 잘못된 추출의 경우가 대부분이었다. DB1의 경우 Temple 테이블의 Creator 필드가 데이터 프로퍼티가 아닌 도메인 온톨로지의 재질 클래스의 토제 클래스와 매칭이 되는 문제점이 있었다. 인스턴스의 수가 많은 경우에는 정확도가 향상될 것으로 기대된다.

VI. 결 론

본 논문에서는 시맨틱 웹 환경에서의 통합 검색을 위한 관계형 데이터베이스와 도메인 온톨로지간의 매칭 방법을 제시하였다. 매칭을 수행하기 위해서 먼저, 관계형 데이터베이스의 메타 정보를 이용하여 데이터베이스의 테이블 정보와 필드 정보, 관계 정보 등을 추출하였고, 추출된 테이블 정보를 이용하여 클래스 후보군을 생성하였다. 또한, 필드의 인스턴스들을 도메인 온톨로지와의 유사도 측정을 통해서 클래스 후보와 데이터 프로퍼티로 분류하였으며, 데이터베이스의 관계 정보를 이용하여 데이터베이스 클래스들간의 오브젝트 프로퍼티로 설정하였다. 최종적으로 추출된 정보들을 온톨로지 구조와 비교하여 매칭 정보를 생성하였다.

향후 제안한 알고리즘을 통해서 매칭한 정보를 이용하여 검색을 수행할 경우 사용자가 질의한 내용과 가장 유사한 정도를 판단하기 위한 알고리즘에 대한 연구가 필요하다.

참 고 문 헌

[1] T. B. Lee, J. Hendler, and O. Lassila, *The Sematic Web*, Scientific Am., pp.34-43, 2001.

[2] N. Shadbolt, T. B. Lee, and W. Hall, "The Semantic Web Revisited", IEEE Intelligent Systems, Vol.21, No.3, pp.96-101, 2006.

[3] D. McGuiness, R. Fikes, J. Hendler, and L. Stein, "DAML+OIL: An Ontology Language for the Semantic Web," IEEE Intelligent Systems, Vol.17, No.5, pp.72-80, 2002.

[4] J. Kang and J. F. Naughton, "On Schema Matching with Opaque Column Names and Data Values," Proceedings of SIGMOD, 2003.

[5] Y. An, A. Borgida, and J. Mylopoulos, "Inferring Complex Semantic Mappings between Relational Tables and Ontologies from Simple Correspondences," Proceedings of ODBASE,

pp.1152-1169, 2005.

[6] G. Javier and D. Riano, "Meta-data and ER Model Automatic Generation from Unstructured Information Resources," 5th Joint Conference on Knowledge-Based Software Engineering, pp.181-186, 2002.

[7] S. R. Upadhyaya and P. S. Kumar, "ERONTO: A Tool for Extracting Ontologies from Extended E/R Diagrams," Proceedings of SAC05, pp.666-670, 2005.

[8] M. Li, X. Y. Du, and S. Wang, "Learning Ontology from Relational Database," Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp.3410-3415, 2005.

[9] Q. Trinh, K. Barker, and R. Alhajj, "RDB2ONT: A Tool for Generating OWL Ontologies from Relational Database System," Proceedings of AICT2006, pp.170-178, 2006.

[10] http://www.museum.go.kr

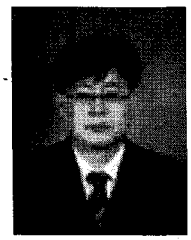
[11] http://www.cha.go.kr

[12] http://www.koreatemple.net

[13] J. D. Bo, P. Spyns, and R. Meersman, "Assisting ontology integration with existing thesauri," Proceedings of ODBASE, pp.801-818, 2004.

저 자 소 개

이 기 정(Ki-Jung Lee) 정회원



- 1999년 2월 : 서울시립대학교 국사학과 학사
- 2003년 2월 : 경원대학교 전자계산학과 석사
- 2004년 3월~현재 : 경원대학교 전자계산학과 박사과정

<관심분야> : 시맨틱 웹, 온톨로지 매칭, 시맨틱 매칭, 내용기반검색

황 보 택 근(Taeg-Keun Whangbo) 정회원

- 1983년 2월 : 고려대학교 공과대학 학사
- 1987년 8월 : CUNY 전산학과 석사
- 1995년 8월 : S.I.T. 전산학과 박사
- 1995년~1997년 : 삼성종합기술원 선임연구원
- 1997년~현재 : 경원대학교 소프트웨어대학 부교수

<관심분야> : 시맨틱 웹, 정보가시화, 내용기반검색