

## 온톨로지 분석 기반의 UML 클래스 모델을 이용한 데이터 통합

서진원<sup>1</sup>, 공헌택<sup>1</sup>, 임재현<sup>1</sup>, 김치수<sup>1\*</sup>

### An Integration of Data by using UML Class Models Based on the Ontology Analysis

Jin-Won Seo<sup>1</sup>, Heon-Tag Kong<sup>1</sup>, Jae-Hyun Lim<sup>1</sup> and Chi-Su Kim<sup>1\*</sup>

**요 약** 데이터 통합은 상이한 소스로부터 이종의 데이터를 결합하고, 사용자에게 단일 접근 점을 통해 다중 소스의 모든 데이터에 투명하게 액세스하는 것을 허용하는 기술이다. 데이터 통합의 어려움은 데이터의 이종성 즉, 스키마의 이종성, 시멘틱의 이종성이다. 따라서 데이터의 풍부한 시멘틱이 데이터 소스의 이종성으로 인한 충돌을 해결하기 위한 주요한 요인이다. 그러나 UML 클래스 모델에서는 데이터의 스키마 기반 시멘틱만을 표현하기 때문에 온톨로지와 같은 대안책을 이용해 추가적인 시멘틱을 제공하는 것이 바람직하다. 본 논문에서는 온톨로지 분석을 이용해 UML 클래스 모델로 표현된 두 데이터 소스를 통합하기 위한 방법을 제시한다. 여기서 온톨로지는 각 소스의 데이터에 대한 시멘틱을 표현하는데 적용된다. 온톨로지의 유사성과 차이점을 결정하기 위해 온톨로지를 분석하고 비교한다. 비교 결과는 통합 정보에 대해 질의가 가능한 통합 온톨로지를 구축하는데 사용된다.

**Abstract** Data integration is techniques to combine heterogeneous data from different sources, and to allow users to transparently access all data from multiple sources via a single view. The difficulty with data integration is data heterogeneity (i.e. schema heterogeneity, semantic heterogeneity). Richer semantics of data is a major factor in resolving conflicts among heterogeneous data sources.

As UML class model represents only schema-based semantics of data, alternative methods such as ontology is useful for representing additional semantics. This paper proposes a method for integrating two data sources with UML class models by using an analysis of their ontologies. In our framework, ontology will be applied to describe semantics of data in each source. Then the ontologies are analysed and compared to determine their similarities and differences. The result of the comparison is used to devise an integrated ontology that will enable querying on the integrated information.

**Key words** : 데이터 통합(Data integration), UML class model(UML 클래스 모델), 온톨로지 분석(Ontology Analysis)

## 1. 서론

정보 시스템에서의 데이터 통합은 상이한 데이터 소스로부터 이종의 데이터를 결합하고, 데이터 접근 권한이 있는 사용자들이 다양한 소스를 단일 접근 방법을 통해 투명하게 접근할 수 있도록 허용하는 기술에 관한 중요한 문제였다.

이를 위해 서로 다른 정보시스템에 있는 데이터들

을 결합하여 통합된 데이터 시스템을 구축함으로써 일관되고 정확한 양질의 데이터를 제공할 수 있다. 그러나 이런 연구들이 갖는 가장 큰 문제점은 스키마의 이종성, 시멘틱의 이종성을 들 수 있으며, 이런 문제가 이종자료의 완벽한 통합에 걸림돌이 되고 있다.

몇몇 연구에서 이런 문제를 해결하기 위해 전역 스키마와 지역 스키마를 두고 전역 스키마와 지역 스키마를 매핑하는 구조를 제시하고 있다. 그러나 이런 방식에서도 서로 다른 이름으로 동일 데이터를 표현하는 이름 불일치, 서로 다른 데이터 타입으로 동일 데이터를 표현하는 구조 불일치, 시멘틱 또는 데이터 레벨 불일치 등이 있을

본 연구는 한국과학재단 특정기초연구 (R01-2006-000-10555-0)지원으로 수행되었음

<sup>1</sup>공주대학교 컴퓨터공학부

\*교신저자 : 김치수(cskim@kongju.ac.kr)

수 있으며, 이러한 예로 다른 범위의 데이터 값, 다른 단위의 데이터 값 등이 있다[1]. 또한 이러한 표현방식은 매핑의 어려움으로 인한 비용과 시간의 문제, 부정확한 매핑의 문제가 발생한다.

UML 클래스 모델의 참조는 스키마 충돌 해결에 어울리는 시멘틱 표현 방법으로 볼 수 있다. 스키마 충돌 이외의 문제에 대한 해결책으로 추가적인 시멘틱을 표현할 수 있는 능력을 가진 온톨로지 개념이 요구되어진다. 그렇지 않으면 데이터 통합자가 데이터 레벨의 충돌을 수작업으로 해결해야 한다.

온톨로지는 어떤 형태의 지식을 표현하고, 개념들 사이의 관계를 나타낼 수 있는 능력을 갖는다[2]. 이것은 스키마 기반 시멘틱과 추가적인 시멘틱을 함께 나타낼 수 있는 공유 어휘를 정의하는 지식표현에 적합하다.

본 논문의 데이터 통합 프로세스는 이종의 데이터에 대한 풍부한 시멘틱을 표현하는 능력, 데이터 통합 알고리즘, 데이터 소스와 관련된 데이터 소스 스키마, 데이터 소스 스키마를 지식 표현으로 변환하는 기술을 포함하고 있다.

본 논문에서는 UML 클래스 모델을 이용한 두 데이터 소스의 통합 과정을 보인다. 지식표현 방법으로 온톨로지를 선택하였고, 온톨로지 통합 알고리즘은 [3]의 객체지향 데이터 모델 통합 방법을 적용하였다.

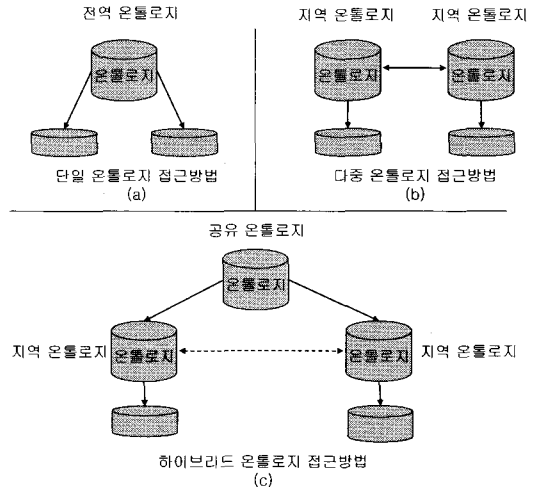
본 논문의 UML 클래스 모델의 데이터 소스 스키마는 OMG에서 제안한 DSTC 온톨로지 정의 모델[4]에 따른 온톨로지 기반 데이터 소스 스키마로 변환된다. 이 온톨로지 기반 스키마는 추가적인 데이터 시멘틱에 의해 확장된다.

OBSERVER[5], FCA-MERGE[6], PROMPT[7], Chimaera[8] 등의 연구에서 이기종의 데이터 통합 기술을 제안하고 있지만 웹 페이지와 같이 비구조적이거나 부분 구조적인 데이터를 통합하는 경향이 있다. 이런 데이터의 본래 성질 때문에 자연 언어 처리의 사용 등으로 알고리즘은 복잡해지게 된다.

본 논문에서는 구조적인 두 UML 클래스 모델을 통합하기 때문에 온톨로지는 보다 더 구조적이고, 제안되는 알고리즘은 덜 복잡하게 된다.

## 2. 관련연구

온톨로지 기반 데이터 통합 방법은 [그림 1]과 같이 세 가지 방법으로 나눌 수 있다[9].



[그림 1] 온톨로지 기반 데이터 통합 방법

### a) 단일 온톨로지 접근방법

SIMS[10]가 대표적인 예로서 전역 온톨로지에 모든 소스의 데이터를 생성하고, 전역 온톨로지를 통해 통합데이터 기반 질의를 요청하는 구조이다. 이 방법은 전역 온톨로지를 정의하기 위해 모든 소스의 데이터에 대하여 잘 아는 전문가를 필요로 한다는 단점이 있다.

### b) 다중 온톨로지 접근방법

OBSERVER[5]가 대표적인 예로서 각각의 소스로부터 지역 온톨로지에 편리하게 데이터를 생성하지만, 지역 온톨로지 사이의 매핑이 이루어져야 한다. 통합 데이터에 대한 질의는 각 지역 온톨로지를 사용하고, 다른 지역 온톨로지에 대한 질의를 구성하기 위해 매핑을 사용한다. 이 방법은 보다 적응력이 있어 보이지만 요구하는 데이터를 제공해줄 수 있는 적합한 매개체로서의 온톨로지를 선택해야 하는 단점이 있다.

### c) 하이브리드 온톨로지 접근방법

COIN[11]과 MECOTA[12]가 대표적인 예로서 지역 온톨로지에 각 소스의 데이터를 생성하고, 공유 온톨로지는 지역 온톨로지 사이의 어휘를 공유하기 위해 사용된다. 지역 온톨로지에 정의하는 용어함과 공유 온톨로지를 통한 질의의 장점을 갖지만, 질의 시에 공유 온톨로지와 각 지역 온톨로지 간의 매핑으로 인해 오버헤드가 발생되는 문제를 갖는다.

따라서 본 논문에서는 단일 온톨로지 방법과 하이브리드 방법을 혼합 사용한 온톨로지 기반의 데이터 통합을 제안한다. 본 논문에서 제시한 방법은 하이브리드 방법에

따라 각 소스의 데이터는 각 지역 온톨로지에 기술하며, 두 지역 온톨로지를 병합하기 위해 통합 온톨로지를 구축한다. 통합 온톨로지는 데이터 소스와 직접 연관되며, 통합된 데이터에 대한 질의는 오직 통합 온톨로지만을 통해 이루어지는 단일 접근 방법을 취한다.

질의 시에 매핑으로 인해 오버헤드가 발생하는 하이브리드 접근방법의 단점을 해결하기 위해 질의는 오직 통합 온톨로지만을 통해 이루어지며, 통합 온톨로지를 지원하기 위해 미리 만들어진 뷰를 통해 각 지역 온톨로지에 대한 질의로 번역하여 전달한다.

[표 1]은 기존의 접근방법들과 본 논문의 접근방법의 장점 및 단점을 보여준다.

또한 UML 클래스 모델에서 온톨로지로의 변환도 많은 연구가 진행되고 있다. [13]에서는 UML, DL, ER과 온톨로지 사이의 매핑을 제안했으며, Java2OWL[14]은 자바 클래스와 사례를 카디널리티 제약과 XML 스키마 데이터타입을 사용하여 OWL로 변환을 시도하였다. 그러나 이런 연구들에서는 데이터의 스키마 기반 시멘틱만을 이용한 온톨로지로의 변환을 다루기 때문에 데이터 통합 시 스키마 충돌 이외의 충돌 문제는 다루지 않고 있다.

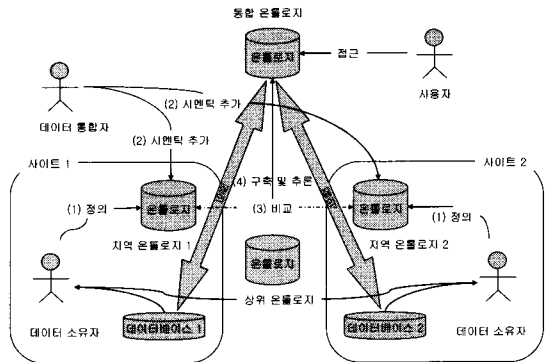
따라서 본 논문에서는 DSTC의 제안을 적용하여 UML 클래스 모델의 데이터 소스 스키마를 온톨로지 기반 데이터 소스 스키마로 변환하며, 이 온톨로지 기반 스키마를 추가적인 데이터 시멘틱에 의해 확장함으로써 풍부한 시멘틱 표현을 갖도록 하여 데이터 통합 시 발생하는 스키마 충돌 이외의 충돌 문제를 해결하고자 한다.

그리고 본 논문에서는 RDF 그래프와 RDF triple이 사용되며, Jena와 RDQL(RDL Data Query Language)이 RDF triple을 위한 추론 엔진과 질의 언어로 사용된다.

### 3. UML 클래스 기반 온톨로지 통합

본 논문의 데이터 통합 방법은 UML 클래스 모델에 이미 존재하는 시멘틱과 다른 추가적인 시멘틱의 표현으로 인한 풍부한 표현력의 장점을 갖는다. 데이터 통합은 [그림 2]와 같이 (1)지역 온톨로지 정의, (2)시멘틱 추가, (3)지역 온톨로지 비교, (4)통합 온톨로지 구축 및 추론의 4 단계로 구성된다.

#### 3.1 지역 온톨로지의 정의

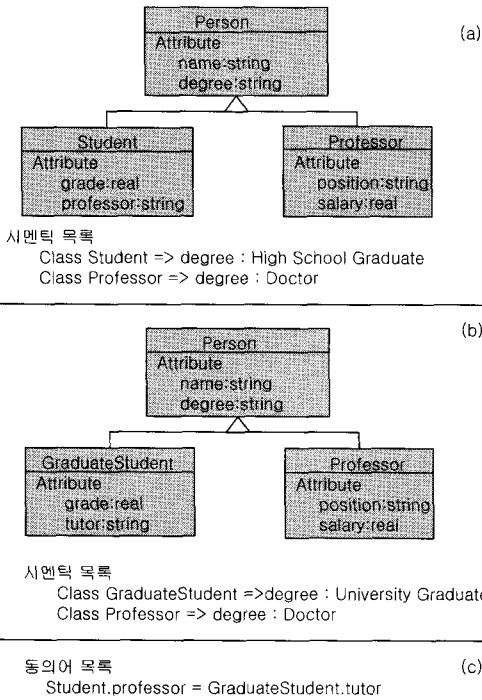


[그림 2] 데이터 통합 과정

[그림 2]의 상위 온톨로지는 두 데이터 모델의 지역 온톨로지에 대한 메타 온톨로지이며, 지역 온톨로지의 데이터 소유자와 데이터 통합자는 상위 온톨로지를 이용하여 시멘틱을 정의하거나 추가한다. 스키마 기반 시멘틱은 UML 클래스 모델에 존재하는 시멘틱으로 [그림 3]의 UML 클래스 다이어그램을 참조하여 클래스와 속성을 나타내는 "Class"와 "ClassAttribute"를 사용한다.

[표 1] 온톨로지 기반 통합방법의 비교

	단일 온톨로지 접근방법	다중 온톨로지 접근방법	하이브리드 온톨로지 접근방법	본 논문의 접근방법
구축 노력	단순하지만 모든 소스에 대해 잘 아는 전문가 필요	많은 노력 필요	적당	적당
의미 이종성	유사한 관점의 단일 온톨로지	이종 관점 제공	이종 관점 제공	이종 관점 제공
소스의 추가/삭제	전역 온톨로지 변경 필요	새로운 소스 온톨로지 제공, 다른 온톨로지와 연결	새로운 소스 온톨로지 제공	새로운 소스 온톨로지 제공
다중 온톨로지 비교	X	공통 어휘 부족으로 비교가 어려움	공통 어휘 사용으로 비교가 용이함	공통 어휘 사용으로 비교가 용이함
질의	전역 온톨로지를 통한 통합데이터 기반 질의	적당한 온톨로지 선택이 요구됨	질의 시 공유 온톨로지와 각 지역 온톨로지 간의 매핑으로 인한 오버헤드 발생	통합 온톨로지를 통한 질의, 매핑으로 인한 오버헤드 문제 해결



[그림 3] UML 클래스 모델과 추가적인 시멘틱

### 3.2 지역 온톨로지의 비교

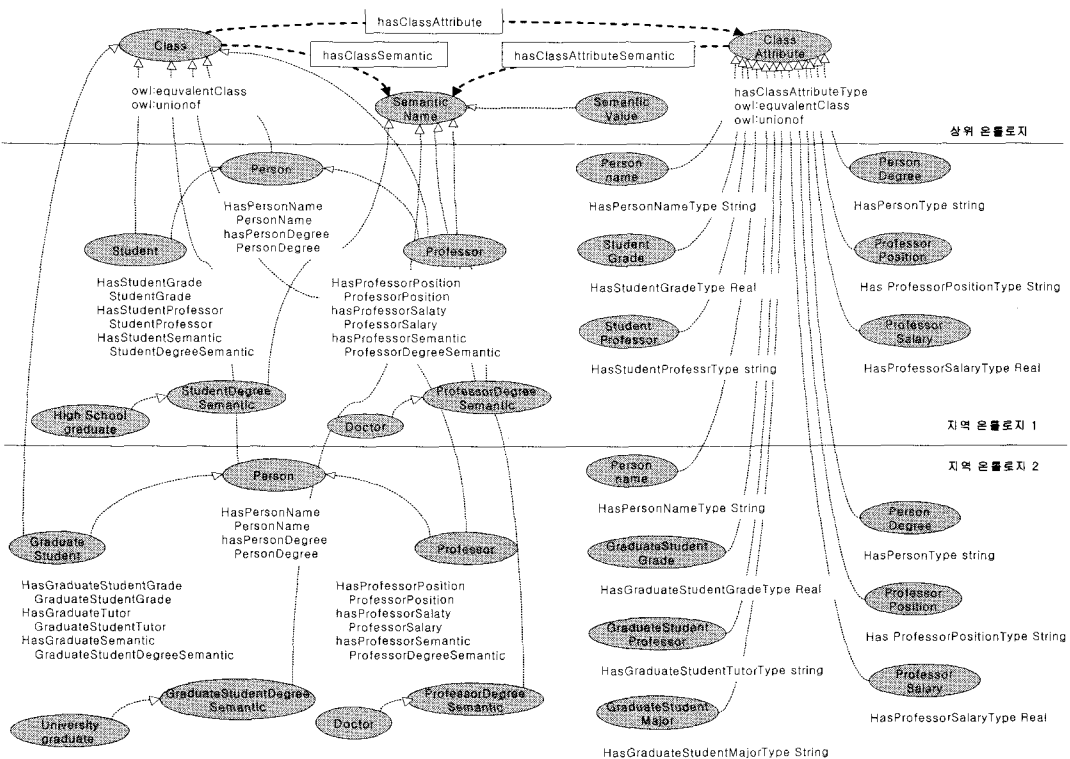
[그림 4]의 상위 온톨로지 "Class"의 서브클래스인 두 UML 클래스 사이의 관계를 발견하기 위해 클래스 비교를 한다. 클래스 비교는 시멘틱 비교와 클래스속성 비교로부터 시작되는 상향식 프로세스이다. 먼저, 두 지역 온톨로지에서 클래스의 각 쌍을 선택한다. 다음으로, 클래스 시멘틱 비교와 클래스 속성 비교를 각각 수행한다. 마지막으로, 클래스 비교에 의해 각각의 클래스 사이의 관계를 발견한다. 자세한 순서 및 방법은 다음과 같다.

#### 3.2.1 시멘틱 비교

시멘틱 비교는 데이터 통합 시스템을 이용한 객체 지향 모델의 데이터 통합을 기반으로 했다[3]. 각각의 시멘틱 이름과 시멘틱 값을 비교할 때 나타날 수 있는 휴리스틱 값  $r$ 을 [표 2]에 보였다.

[표 2] 시멘틱 T1과 T2의 비교 결과 값  $r$

Case	시멘틱 이름	시멘틱 값	$r$
1	동일	동일	1
2	동일	서브셋	0.8
3	동일	겹침	0.8
4	동일	연결 없음	0.5
5	다름	전부 다름	0



[그림 4] 상위 온톨로지와 지역 온톨로지

T1은 지역 온톨로지 1의 항목이고, T2는 지역 온톨로지 2의 항목이다.  $\alpha$ 는 T1의 시멘틱 이름의 수,  $\beta$ 는 T2의 시멘틱 이름의 수이며  $\alpha \leq \beta$ 이다. 시멘틱 쌍들의 비교 값  $R=\{r1, r2, \dots, rm\}$ 이 반영된 휴리스틱 점수  $Ms=\sum ri, ri \in R,$

( $i=1, \dots, n$ )이다. [표 3]은 T1이 T2와 이룰 수 있는 관계의 종류와 s 값이다.

[표 3] 시멘틱을 고려한 관계

Case	Ms	결과	관계	s
1	$\alpha(\alpha = \beta)$	동치	동치	1
2	$\alpha(\alpha < \beta)$	동치	슈퍼클래스*	1
3	$\geq \alpha(\alpha = \beta)$	동치	동치	1
4	$\geq \alpha(\alpha < \beta)$	포함	슈퍼클래스*	1
5	$\geq \alpha$	강한 집합	형제관계	0.1
6	$< \alpha$	약한 집합	연결 없음	0
7	0	관계 없음	연결 없음	0

\*T1은 T2의 슈퍼클래스=T2는 T1의 서브클래스

3.2.2 클래스 속성 비교

각 쌍의 클래스 속성을 비교하여 [표 4]에 따른 휴리스틱 값 s를 기록한다. 휴리스틱 값은 비교되는 두 클래스 사이의 관계 정도를 말한다.

[표 4] 클래스 속성 A와 B의 비교 결과에 따른 관계

Case	비교 조건	s	관계
1	A의 이름이 B와 같고 추가적인 시멘틱이 정의됨	표2 참조	표2 참조
2	A의 이름이 B와 다르지만 동치, 추가적인 시멘틱 정의됨	표2 참조	표2 참조
3	A의 이름이 B와 같고 추가적인 시멘틱 정의되지 않음	1	동치
4	A의 이름이 B와 다르지만 동치, 추가적인 시멘틱 정의되지 않음	1	동의어
5	A unionOf B 또는 B unionOf A	1	하위어
6	A가 B와 다름	0	연결없음

[표 5]는 클래스 속성 집합을 고려한 C1과 C2의 관계의 종류이다. C1과 C2는 비교되는 지역 온톨로지 1과 지역 온톨로지 2의 클래스 속성이다.  $\theta c1$ 은 C1의 클래스 속성의 수,  $\theta c2$ 는 C2의 클래스 속성의 수이며,  $\alpha = \theta c1,$   $\beta = \theta c2,$   $\alpha \leq \beta$ 이다. C1의 클래스 속성이 C2의 클래스

속성과 “owl:unionOf” 프로퍼티를 갖는다면  $\alpha = \alpha - 1 + C1$ 의 클래스 속성 수, C2의 클래스 속성이 C1의 클래스 속성과 “owl:unionOf” 프로퍼티를 갖는다면  $\beta = \beta - 1 + C2$ 의 클래스 속성 수이다.

클래스 속성의 비교 결과 값의 집합은  $S=\{s1, s2, \dots, sn\}$ 이고,  $Ms=\sum si, si \in S, (i=1, \dots, n)$ 로 계산한다.

[표 5] 클래스 속성을 고려한 C1과 C2의 관계

Case	Ms	결과	관계
1	$\alpha(\alpha = \beta)$	동치	동치
2	$\alpha(\alpha < \beta)$	포함	슈퍼클래스*
3	$\geq \alpha(\alpha = \beta)$	동치	동치
4	$\geq \alpha(\alpha < \beta)$	포함	슈퍼클래스*
5	$\geq \alpha$	강한 결합	형제관계
6	$< \alpha$	약한 결합	연결 없음
7	0	관계없음	연결 없음

\*C1은 C2의 슈퍼클래스=C2는 C1의 서브클래스

3.2.3 클래스 비교

클래스 C1과 C2의 관계는 3.2.1과 3.2.2에 보인 클래스 시멘틱 관계와 모든 클래스 속성 사이의 관계에 기초한다. 비교 결과 가질 수 있는 관계는 [표 6]과 같다.

[표 7]은 [그림 4]의 클래스 ‘Student’와 ‘GraduateStudent’의 비교를 보여준다. 비교는 3.2.1, 3.2.2, 3.2.3의 순서를 따른다. ‘Student’가 ‘GraduateStudent’의 슈퍼클래스인 클래스 시멘틱과 클래스 속성 집합을 갖는다. 따라서 ‘Student’는 ‘GraduateStudent’의 슈퍼클래스이다. 두 온톨로지의 다른 클래스의 쌍도 같은 방법으로 비교된다.

[표 6] 클래스 관계 요약

Case	클래스 시멘틱 비교 결과	클래스 속성 비교 결과	관계
1	- 동치 - 슈퍼클래스 - 서브클래스 - 정의 안 됨	동치	동치
2	- 동치 - 슈퍼클래스 - 서브클래스 - 정의 안 됨	슈퍼클래스	슈퍼클래스
3	- 동치 - 슈퍼클래스 - 서브클래스 - 정의 안 됨	서브클래스	서브클래스
4	- 동치 - 슈퍼클래스 - 서브클래스 - 정의 안 됨	형제관계	형제관계
5	- 연결 없음	형제관계	형제관계

[표 7] 'Student' 클래스와 'GraduateStudent' 클래스의 비교

시맨틱 비교	
Ontology2.GraduateStudent $\subset$ Ontology1.Student => Ontology2.GraduateStudent.Age는 Ontology1.Student.Age의 서브셋 $\therefore R=[0.8], M_s = 0.8, \alpha = 1, \beta = 1$ , 결과=동치, 관계=동치	표 2의 Case2 표 3의 Case3
클래스 속성 비교	
Ontology1.Student.PersonName은 Ontology2.GraduateStudent.PersonName과 같고 추가적인 시맨틱 정의되지 않음 $\Rightarrow$ 관계=동치, $s=1$	표 4의 Case3
Ontology1.Student.PersonAge는 Ontology2.GraduateStudent.PersonAge와 같고, Ontology2.GraduateStudent.PersonAge의 시맨틱 $\subset$ Ontology1.Student.PersonAge의 시맨틱 $\Rightarrow$ 결과=슈퍼클래스, $s=1$	표 4의 Case1 표 3의 Case3
Ontology1.Student.StudentGrade는 Ontology2.GraduateStudent.GraduateStudentGrade와 같지만, 추가적인 시맨틱 없음 $\Rightarrow$ 결과=동치, $s=1$	표 4의 Case3
Ontology1.Student.Professor는 Ontology2.GraduateStudent.GraduateStudentTutor와 다르지만 동치, 추가적인 시맨틱 없음 $\Rightarrow$ 결과=동치어, $s=1$	표 4의 Case4
$\therefore M_s=1+1+1+1=4, \alpha = 4, \beta = 5$ 관계=슈퍼클래스	표 5의 Case2
클래스 비교	
$\therefore$ 표5에서 클래스 시맨틱 관계 + 클래스 속성 관계 = 동치 + 슈퍼클래스 = 슈퍼클래스	표 6의 Case2

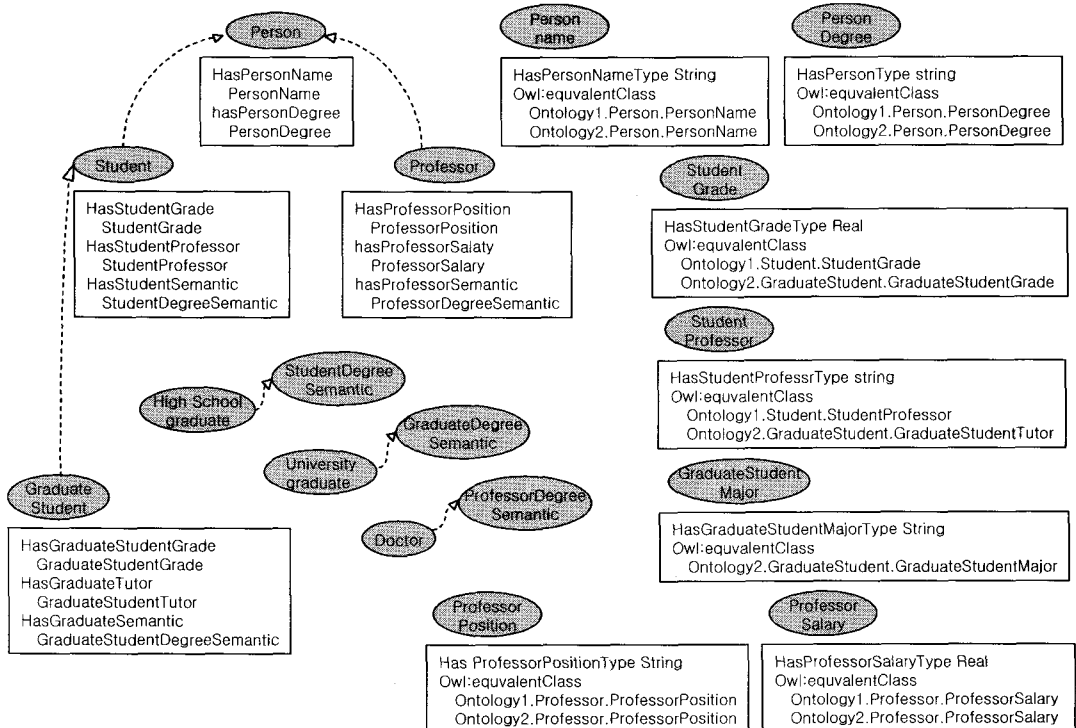
#### 4. 데이터 소스에 통합 온톨로지 적용

온톨로지 비교 결과는 각 클래스 쌍들을 비교하여 동치, 슈퍼클래스, 서브클래스, 형제관계와 같이 관계를 식별한다. 관계 결과에 의해 새로운 통합 온톨로지를 구축할 수 있다. [표 8]은 클래스 쌍들의 비교 결과이다.

[표 8] 온톨로지 1과 온톨로지 2의 비교 결과

	Ontology1.Person	Ontology1.Student	Ontology1.Professor
Ontology2.Person	동치	슈퍼클래스	슈퍼클래스
Ontology2.GraduateStudent	서브클래스	서브클래스	형제관계
Ontology2.Professor	서브클래스	형제관계	동치

비교한 클래스 각 쌍들의 관계 결과에 따라 통합의 순서가 결정된다. 온톨로지 내의 클래스가 다른 온톨로지 내의 클래스와 통합되기 전에 반드시 클래스와 다른 온톨로지 내의 클래스 사이의 관계를 결정해야 한다. 동치 관계가 가장 높은 우선순위를 갖으며 가장 먼저 통합되고 슈퍼클래스, 서브클래스, 형제관계 순으로 통합이 이루어진다.



[그림 5] 통합 온톨로지

[그림 5]는 'Person'의 서브클래스인 'Student'와 'Professor'는 루트 클래스인 'Person'으로부터 상속받고, 'GraduateStudent'도 'Student'로부터 상속받는 통합 온톨로지이다.

통합 온톨로지에 추론 규칙을 추가하면 질의 처리 시에 온톨로지의 능력과 정확성을 증가시켜 준다. 'Student'와 'GraduateStudent'의 시멘틱에 “대학 졸업자 이상의 'Student'는 'GraduateStudent'이다.” 라는 사실을 추가할 수 있다.

통합 온톨로지가 만족스럽지 못하면 온톨로지에 더 많은 시멘틱을 추가하기 위해 시멘틱 추가 단계로 되돌아가게 된다.

통합 온톨로지가 생성되고, 추론이 되면 통합 온톨로지의 모든 결과 클래스는 지역 온톨로지에 매핑 된다. 질의는 통합 온톨로지를 지나서 지역 온톨로지의 데이터 소스에 대한 질의로 변환된다.

본 논문에서 제시한 질의의 경로와 변환을 보이기 위해 두 개의 질의 예를 보이고자 한다. 질의는 RDQL을 따르며, RDF triple에서 정보를 추출하기에 적합하다. 질의가 통합 온톨로지에 전달되면 통합 온톨로지는 모든 데이터 소스의 DBMS에 RDQL을 보낸다. 그다음 통합 온톨로지의 지원을 위해 미리 만들어진 뷰는 RDQL을 다른 DBMS 질의어로 번역하는 역할을 한다.

예제 1

```
SELECT ?ProfessorPosition, ?ProfessorSalary
WHERE (?ProfessorPosition info:position
?ProfessorSalary) AND ?Professor <= 3000000
```

예제 1의 질의는 통합 온톨로지로부터 3000000 이하의 급여를 받는 Professor를 직위와 함께 찾기 원한다. 이 질의는 두 데이터 소스의 DBMS에 넘어갈 것이다. 이 질의는 각각 지역 온톨로지1과 지역 온톨로지2에 의해 기술되는 데이터 소스1과 2에서 다음과 같은 SQL 질의로 번역될 것이다.

```
SELECT Professor.position, Professor.salary
FROM Professor
WHERE Professor.salary <= 3000000
```

예제 2

```
SELECT ?StudentName, ?StudentGrade
WHERE (?StudentName info:grade ?StudentGrade)
AND ?StudentGrade > 4.0
```

예제 2의 질의는 통합 온톨로지로부터 평점이 4.0 이상인 학생의 이름과 학점을 찾기 원한다. 이 질의는 두 데이터 소스의 DBMS에 넘어갈 것이다. 이 질의는 데이터 소스 1에서 다음과 같은 SQL 질의로 번역될 것이다.

```
SELECT Student.name, Student.grade
FROM Student
WHERE Student.grade > 4.0
```

이 질의는 데이터 소스2에서는 다음과 같은 SQL 질의로 변환될 것이다.

```
SELECT GraduateStudent.name,
GraduateStudent.grade
FROM GraduateStudent
WHERE GraduateStudent.grade > 4.0
```

### 5. 결론 및 향후 연구

UML 클래스 모델과 같은 구조적인 데이터를 통합하기 위한 온톨로지의 사용은 온톨로지의 풍부한 표현력의 장점으로 인해 좋은 통합 결과를 얻을 수 있다. 질의는 통합 온톨로지를 통해 데이터 소스에 대한 질의로 변환된다. 질의가 온톨로지의 추론 능력에 의한 시멘틱을 바탕으로 하기 때문에 사용자는 온톨로지를 이용해 시멘틱을 기술함으로써 이익을 얻을 수 있다. 본 논문에서는 온톨로지 분석 기반 UML 클래스 모델 통합 방법을 제시하였다. UML 클래스 모델과 같이 구조적인 데이터의 통합은 데이터의 스키마 기반 시멘틱만을 표현한다. 스키마 충돌 이외의 문제에 대한 해결책으로 추가적인 시멘틱을 표현할 수 있는 능력을 가진 온톨로지 개념이 적용되었다.

향후 연구로는 온톨로지 분석 기반 UML 클래스 모델 통합 방법을 정제하고, 속성에 대한 제약조건과 같은 더 많은 종류의 시멘틱을 지역 온톨로지에 포함시킬 것이다. 또한 실무 어플리케이션의 데이터 모델에 적용하고, 다른 연구들과 비교하여 성능을 평가하고자 한다.

### 참고문헌

[1] Ram, S. and Park, J. Semantic conflict resolution ontology(SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts. Knowledge and Data Engineering, IEEE Transactions on, 16 (2). 189-202, 2004.

[2] Gruber, T., A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5 (2). 199 - 220, 1993.

[3] Swasrukkiet, S. A Methodology for Integration of Object-Oriented Data Models Using Heuristics and Analysis of Relationships between Class. Department of Computer Engineering, Chulalongkorn University, Bangkok, 1999.

[4] DSTC. Ontology Definition MetaModel - DSTC Initial Submission, OMG, 2003.

[5] Mena, E., Illarramendi, A., et al. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. Distributed and Parallel Databases -- An International Journal, 8 (2). 223-271, 1996.

[6] Stumme, G. and Maedche, A., FCA-MERGE: Bottom-Up Merging of Ontologies. In The Seventeenth International Joint Conference on Artificial Intelligence, 2001.

[7] Noy, N.F. and Musen, M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. in The Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, (2000), AAAI Press / The MIT Press, 450 - 455.

[8] McGuinness, D.L., Fikes, R., et al., An Environment for Merging and Testing Large Ontologies. in Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), (Breckenridge, Colorado, 2000).

[9] Wache, H., Vogele, T., et al., Ontology-Based Integration of Information - A Survey of Existing Approaches. in The Seventeenth International Joint Conference on Artificial Intelligence, 2001.

[10] Arens, Y., Hsu, C.-N., et al. Query Processing in the SIMS Information Mediator. in cadings in agents, Morgan Kaufmann Publishers Inc, 82 - 90, 1998.

[11] Goh, C.H., Bressan, S.p., et al. Context Interchange: New Features and Formalisms for the Intelligent Integration of Information. ACM Transactions on Information Systems (TOIS), 17 (3). 270-293, 1997.

[12] Wache, H., Scholz, T., et al., An integration method for the specification of rule-oriented mediators. in Database Applications in Non-Traditional Environments, 1999. (DANTE '99) Proceedings. 1999 International Symposium on, 109-112, 1999.

[13] Colomb, R.M., Gerber, A., et al., Issues in Mapping Metamodels in the Ontology Development Metamodel

Using QVT. in The 1st International Workshop on the Model-Driven Semantic Web (MSDW 2004), 2004.

[14] Dean, M. Java2OWL, DAML.org, 2003.

**서진원 (Jin-Won Seo)**

[정회원]



- 1994년 한밭대학교 전자계산학과 졸업(학사)
- 2004년 공주대학교 멀티미디어공학과 졸업(석사)
- 2008년 현재 공주대학교 컴퓨터공학과 박사과정

<관심분야>  
데이터통합, UML, 온톨로지

**공헌택 (Heon-Tag Kong)**

[정회원]



- 1984년 Northeast Missouri State Univ. 전산학과(학사)
- 1987년 Utah State Univ. 전산학과(석사)
- 1998년 단국대학교 전산통계학과(박사)
- 1988년~ 1990년 한국국방연구원 전산체계연구부 근무
- 1990년~ 현재 공주대학교 컴퓨터공학부 교수

<관심분야>  
병렬 알고리즘, 병렬처리 컴퓨터, 데이터베이스

**임재현 (Jac-Hyun Lim)**

[정회원]



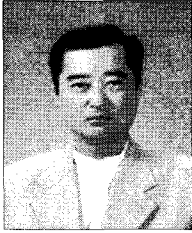
- 1986년 중앙대학교 전자계산학과 졸업(학사)
- 1988년 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1998년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
- 1998년 ~ 현재 공주대학교 컴퓨터공학부 부교수

<관심분야>  
상황인식, RFID/USN, 온톨로지, 인터넷 기술



김 치 수 (Chi-Su Kim)

[정회원]



- 1984년 중앙대학교 전자계산학과 졸업(학사)
- 1986년 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1990년 중앙대학교 대학원 전자계산학과 졸업(박사)
- 1990.9.~1992.8 공주교육대학교 전임강사

• 1992년 ~ 현재 공주대학교 컴퓨터공학부 교수

<관심분야>

소프트웨어 개발 방법론, 온톨로지