

XPOS: 효율적인 질의 처리를 위한 XPath 기반의 OWL 저장 모델

(XPOS: XPath-based OWL Storage Model for Effective Query Processing)

김진형[†] 정동원^{**} 백두권^{***}
(Jinhyung Kim) (Dongwon Jeong) (Doo-Kwon Baik)

요약 최근 인터넷의 급속한 발달로 인해 웹 상의 정보 양이 엄청나게 증가하고 있다. 하지만 현재 웹 상의 정보들은 사람이 읽고 해석하기에만 편리하게 설계되어 원하는 정보에 대한 정확한 검색이 어려워지고 있다. 이러한 문제점을 해결하기 위해 시맨틱 웹이 제안되었으며, 이를 구축하기 위해서는 데이터에 의미를 부여하는 온톨로지 언어를 사용해야 한다. 대표적인 온톨로지 언어는 RDF, RDF-S, OWL 등이 있다. 이러한 언어들 중 OWL은 W3C에서 권고안으로 채택한 언어로써 다른 온톨로지 언어에 비해 풍부한 표현력과 형식적인 의미론을 지닌다. 또한 OWL 데이터는 클래스/프로퍼티들 간의 계층 구조 정보를 포함한다. 따라서, 시맨틱 웹 상에서 데이터의 효율적인 검색을 위해서는 계층적 구조를 고려한 효율적인 OWL 저장 모델이 필요하다. 이 논문에서는 OWL 데이터의 클래스/프로퍼티들 간 계층 정보를 XPath 형태로 포함하며, 직관적이고 효율적인 질의 처리가 가능한 저장 모델(XPOS Model) 및 시스템을 제안한다. 또한, 제안 모델, Sesame, XML 저장소 기반 저장 시스템을 질의 처리 측면에서 성능에 대한 비교 평가 결과를 보인다.

키워드 : XPath, OWL, 계층구조, 저장소 모델, 시맨틱 웹

Abstract With rapid growth of Internet, the amount of information in the Web is increasing exponentially. However, information on the current Web is understandable only for human, and thus it makes the exact information retrieval difficult. For solving this problem, the Semantic Web is suggested and we must use ontology languages that can endow data to semantics for implementing it. One of the representative ontology languages is OWL (Web Ontology Language) adopted as a recommendation by the World-Wide Web Consortium. OWL has richer expression power and formal semantics than other ontology languages such as RDF and RDF-S. In addition, OWL includes hierarchical structure information between classes or properties. Therefore, an efficient OWL storage model considering hierarchical structure for effective information retrieval on the Semantic Web is required. In this paper, we suggest the XPOS (XPath-based OWL Storage) model including hierarchy information between classes or properties as XPath form and enabling intuitive and effective information retrieval. Also, we show the comparative evaluation results on the performance of XPOS model, Sesame, and the XML storage-based storage model regarding query processing.

Key words : XPath, OWL, Hierarchical structure, Storage Model, Semantic Web

[†] 학생회원 : 고려대학교 컴퓨터학과

koolmania2006@gmail.com

^{**} 종신회원 : 군산대학교 정보통계학과 교수

djeong@kunsan.ac.kr

^{***} 종신회원 : 고려대학교 컴퓨터학과 교수

baikdk@korea.ac.kr

논문접수 : 2007년 8월 21일

심사완료 : 2008년 1월 18일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이터베이스 제35권 제3호(2008.6)

1. 서론

최근 웹이 빠른 속도로 발전함에 따라 방대한 양의 웹 정보가 생성 및 유통되고 있다. 하지만 웹 상 정보의 양이 증가함에 따라 원하는 정확한 정보에 대한 검색 작업은 더욱 어려워지고 있다. 즉, 데이터를 얼마나 빠르게 처리하는지 보다 원하는 정보를 얼마나 정확하게 획득하는가에 대한 문제가 중요한 이슈로 대두되고 있다. 이러한 문제점은 현재 웹 상 수많은 데이터들이 사람이 읽고 이해하기 쉽도록 구성되었기 때문에 발생한다. 따라서 이런 문제점을 해결하기 위해 제안된 것이 시맨틱 웹이다[1-3].

시맨틱 웹(Semantic Web)은 사람이 읽고 해석하기에 편리하게 설계되어 있는 현재 웹의 문제점을 해결하고 컴퓨터가 이해할 수 있는 형태의 새로운 언어로 표현하여 기계들 간 상호 의사소통을 할 수 있는 지능형 웹이다. 원리는 사람들이 이해할 수 있도록 자연어 위주로 되어 있는 현재의 웹 문서 기술언어와 달리 정보자원들 사이에 연결되어 있는 의미를 컴퓨터가 이해할 수 있는 형태의 언어로 바꾸는 것이다. 따라서 정보자원의 뜻을 해석하고 기계들끼리 서로 정보를 주고받으면서 자체적으로 필요한 일을 처리하는 것이 가능해진다. 이러한 시맨틱 웹 구축을 위해서는 정보의 의미를 형식적으로 기술할 수 있는 온톨로지 언어를 이용해야 한다. RDF[4-6]는 W3C에서 메타데이터를 기술하기 위해 제정한 표준으로 기존의 HTML이나 XML이 가지고 있는 장점과 더불어 메타데이터를 기술할 수 있는 장점을 지닌다[7-9]. OWL[10,11]은 가장 최근에 W3C 권고안으로 채택된 온톨로지 언어로서 어휘를 구성하는 용어의 의미와 용어들 간 관계를 명시적으로 표현할 수 있다. 또한 OWL은 풍부한 어휘와 형식적 의미론을 포함하고 있기 때문에 XML, RDF, RDF-S보다 풍부한 표현력을 제공하며 추론 기능을 제공한다.

시맨틱 웹 상에서 정확한 정보 검색을 위해서는 OWL 데이터를 효율적으로 저장하는 방법이 필요하다. OWL 데이터의 효율적인 저장을 위해서는 클래스와 프로퍼티에 대한 계층 구조를 고려해야 한다. 현재까지 Sesame[12]와 XML DB 기반 시스템[13]이 OWL 데이터 저장 시 계층 구조에 대해 고려하고 있다. 하지만 Sesame는 계층 구조가 깊어질수록 구조 검색의 수많은 반복으로 인해 비효율적이며, [13]에서 제안한 XML DB 기반 시스템은 매 질의 시 XML 저장소와 RDB에 동시에 질의하고 결과를 다시 통합하는 과정이 반복되어야 하므로 비효율적이다.

이 논문에서는 OWL 데이터 구조를 모두 고려하며 기존 방법에 비해 보다 효율적인 계층 구조 추출이 가

능한 저장 기법을 제안한다. 즉, Sesame처럼 OWL 문서의 모든 데이터를 RDB에 함께 저장하되 계층 구조를 XPath 형태로 저장함으로써 계층 구조에 대한 질의 시 상위 클래스/프로퍼티와 하위 클래스/프로퍼티의 관계에 대한 반복 검색이 발생하지 않고 효율적인 검색이 가능한 저장 구조를 제안한다.

이 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로서 계층 구조를 고려한 OWL 저장 시스템인 Sesame와 [13]에서 제안한 XML 저장소 기반의 저장 시스템에 대해 기술하고, 제 3장에서는 논문에서 제안하는 효율적인 OWL 저장 모델인 XPOS(XPath-based OWL Storage) 모델과 OWL 문서를 XPOS 모델에 변환, 저장하기 위한 XPOS 시스템의 아키텍처 및 세부 모듈에 대해 기술한다. 또한 실제 예제를 통한 XPOS 시스템의 동작 과정 및 중간 데이터, 그리고 최종적으로 XPOS 모델에 저장된 데이터를 상세히 기술한다. 제 5장에서는 성능평가를 통해 기존의 모델(Sesame, XML 저장소 기반 저장 모델)들과 XPOS 모델을 비교 평가한다. 마지막으로 제 6장에서는 결론과 향후 연구에 대해 기술한다.

2. 관련 연구

2.1 Sesame

Sesame[12]는 IST 프로젝트인 On-To-Knowledge의 일환으로 개발된 시스템으로 RDF와 RDF-S를 지원하는 온톨로지 저장, 검색 및 추론 기능을 제공하는 시스템이다. Sesame는 온톨로지 저장을 위해 대표적으로 class, property, resources, subclassOf, subPropertyOf, TRIPLES의 6개의 테이블을 구성하며 이 외에도 domain, range, namespaces, type, labels, comment, literals 테이블로 구성된다. 각 테이블의 간략한 내용은 표 1과 같다.

표 1 Sesame 저장 구조 개요

테이블 명	포함 내용
class	온톨로지 내의 클래스 정보
property	온톨로지 내의 프로퍼티 정보
namespace	온톨로지가 포함되는 네임스페이스 정보
resources	네임스페이스와 클래스 로컬 이름의 자원 정보
range	클래스/프로퍼티의 해당
domain	클래스/프로퍼티의 해당 도메인 정보
type	클래스와 해당 네임스페이스 관계 정보
lables	리터럴과 해당 네임스페이스 관계 정보
triples	클래스/인스턴스/프로퍼티 간의 관계 정보
comment	리터럴과 해당 네임스페이스 관계에 대한 코멘트 정보
literals	클래스에 해당하는 값 정보
subclassOf	클래스 간 계층 구조 정보
subPropertyOf	프로퍼티 간 계층 구조 정보

class 테이블은 온톨로지 문서 내 클래스들에 대한 정보를 포함한다. 클래스간 계층 정보에 무관하게 존재하는 모든 테이블에 대한 이름을 저장한다. property 테이블은 온톨로지 문서 내 프로퍼티들에 대한 정보를 포함한다. class 테이블과 마찬가지로 프로퍼티들간 계층 관계에 무관하게 모든 프로퍼티에 대한 이름을 저장한다. literal 테이블은 각 네임스페이스 및 클래스 해당하는 인스턴스(value)들에 대한 정보를 포함한다. 인스턴스 식별자를 저장하는 id 속성과 인스턴스 값을 저장하는 value 속성으로 구성된다. subClassOf 테이블은 클래스들간 계층 정보를 포함한다. 부모 클래스 이름을 저장하는 superclass 속성과 자식 클래스 이름을 저장하는 subclass 속성으로 구성된다. subPropertyOf 테이블은 프로퍼티들간 계층 정보를 포함하며, 부모 프로퍼티 이름을 저장하는 superprop 속성과 자식 프로퍼티 이름을 저장하는 subprop 속성으로 구성된다. subClassOf와 subPropertyOf 테이블은 인접한 부모/자식 클래스/프로퍼티의 정보만을 포함한다. TRIPLES 테이블은 클래스 또는 클래스 인스턴스와 프로퍼티의 관계 정보를 포함하며, subject-predicate-object의 트리플 구조로 저장된다. 그림 1은 Sesame의 저장 테이블 구조를 보여준다.

하지만, Sesame에서 계층 구조 추출 방법은 계층 구조가 깊고 복잡해질수록 하위 구조를 검색을 위한 작업이 반복되어야 하므로 비효율적이다. 즉, 클래스나 프로퍼티의 계층 구조를 추출하기 위해서는 subClassOf 테이블이나 subPropertyOf 테이블로부터 특정 클래스나

표 2 Sesame의 클래스 간 계층 관계

Superclass Superproperty	Subclass Subproperty
A	B
B	C
C	D
D	E
E	F
F	G
G	H
I	J
J	K

프로퍼티 이름을 상위 속성으로부터 검색한 후, 이 튜플의 하위 속성을 다시 상위 속성으로 갖는 속성 탐색 작업을 더 이상 하위 정보가 존재하지 않을 때까지 반복해야 한다. 예를 들어, 표 2와 같은 경우, 클래스 A와 클래스 K의 계층 구조를 추출하기 위해서는 A와 B의 관계부터 J와 K의 관계까지 총 10회의 반복적인 작업을 거친 후에 클래스 A는 클래스 K의 Super 클래스라는 것을 알 수 있다. 따라서 계층 구조가 복잡하고 깊이가 깊을수록 클래스/프로퍼티간 계층 정보 추출 시간이 증가한다.

2.2 XML 저장소 기반의 OWL 저장 시스템

XML 저장소 기반 OWL 저장 시스템은 OWL 데이터 내 계층 정보에 대한 효율적인 저장 및 검색을 위한 시스템으로 계층 구조 정보는 XML 저장소에 저장하고, Class, Property, Instance 정보는 RDB에 저장한다[13]. 또한, XML 저장소 기반 OWL 저장 시스템은 클래스와

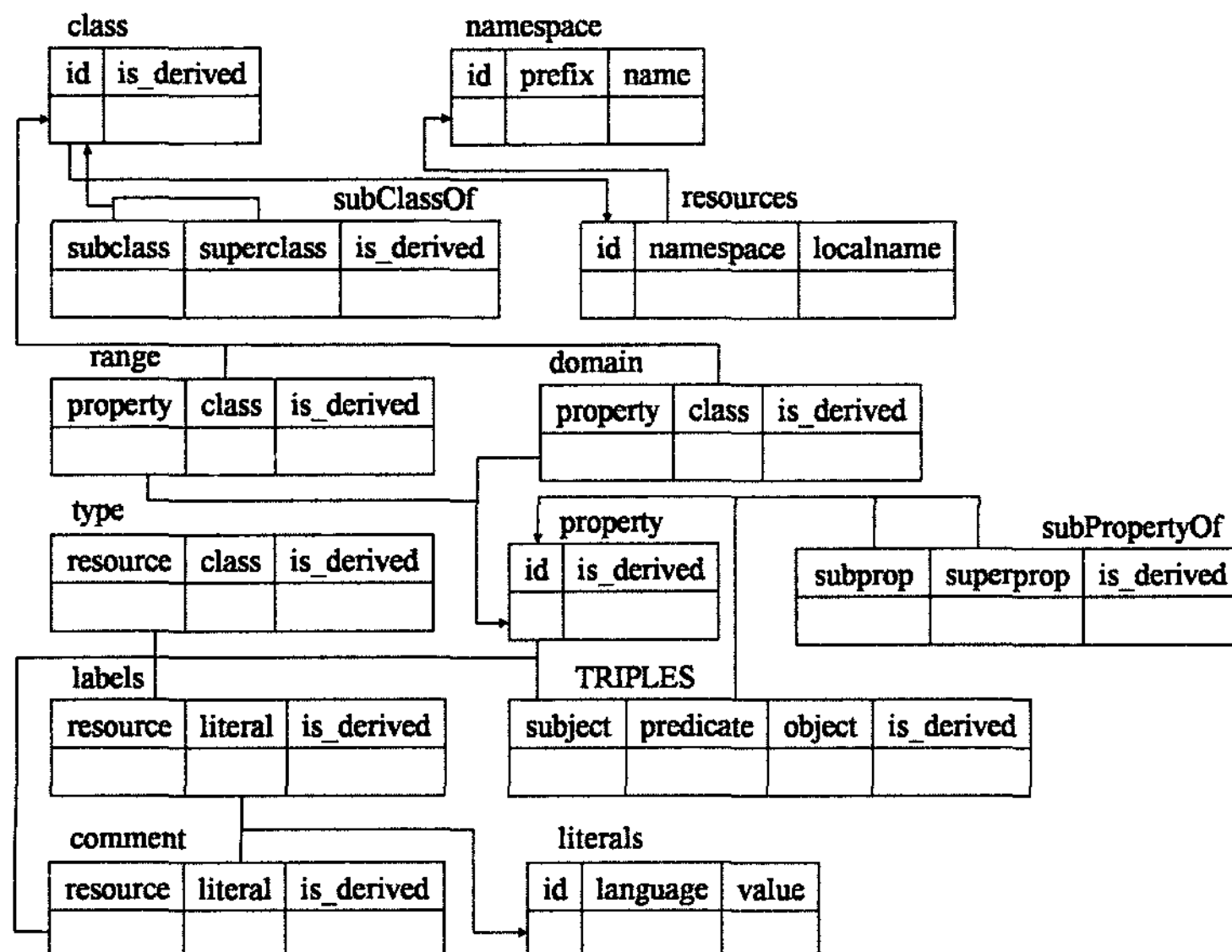


그림 1 Sesame의 저장 테이블 구조

프로퍼티에 대한 계층 정보를 유지하기 위해서 추가적으로 XML 문서를 생성하고 이 문서를 XML 저장소에 저장한다. 즉, OWL 문서의 스키마 부분에 선언되어 있는 클래스와 프로퍼티 정보들 중에 상속 관계를 가지는 클래스와 프로퍼티의 고유한 ID를 가지고 간단한 XML 문서를 생성한다.

그림 2는 OWL 데이터를 저장하기 위한 저장 처리기 구조를 보여준다. OWL 문서가 입력으로 들어오면 OWL 파서가 문서를 파싱해서 클래스, 프로퍼티, 그리고 인스턴스에 관한 정보와 계층 정보, 프로퍼티에 대한 제약 조건 등을 추출해 낸다. 이 때 클래스와 프로퍼티에 대한 계층 정보는 계층 정보 관리자에 의해 XML 문서로 변환된다. OWL 데이터의 계층 정보를 포함한 XML 문서는 XML 저장소에 저장된다. OWL 파서를 통해서 추출된 나머지 정보들은 온톨로지 관리자를 통해서 RDBMS에 저장된다. 그림 3은 계층 정보 저장을 위해 생성된 XML 문서를 나타낸다.

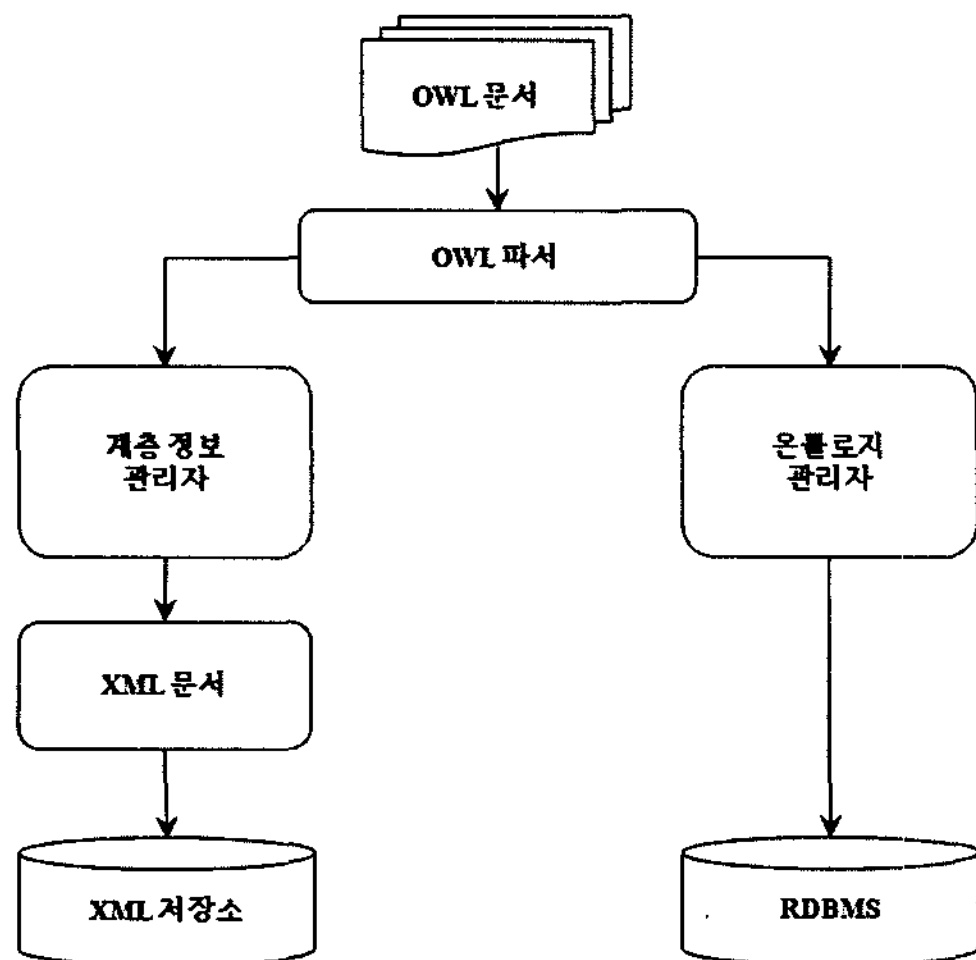


그림 2 XML 저장소 기반의 OWL 저장 시스템의 저장 시스템

XML 저장소 기반 OWL 저장 시스템의 경우, 질의 처리 시마다 XML 저장소에 접근하여 관련 클래스/프로퍼티의 계층 정보를 획득한 후 이 정보를 토대로 RDBMS에 질의하여 최종 결과를 도출해낸다. 즉, 매 질의 마다 XML 저장소와 RDBMS에 모두 접속해야 하며 사용자의 최초 질의를 XML 저장소에 적절한 XQuery와 RDBMS에 적절한 SQL 질의로 변환해야 하므로 성능 면에서 매우 비효율적이다. 또한 특정 클래스에 대한 인스턴스 검색 시 효율성을 높이기 위하여 모든 인스턴스를 하나의 클래스에 저장하지 않고 Sesame와 마찬가지로 각 최상위 클래스 별로 테이블을 생성하여 관련 인스턴스 정보를 저장한다. 따라서 인스턴스 정보가 증가하는 경우 특정 클래스에 대한 인스턴스를 좀 더 빠르게 검색 가능하다. 하지만, 인스턴스가 클래스/프로퍼티 별로 별도 저장된 경우 모든 인스턴스 정보에 대한 질의나 많은 클래스들의 인스턴스 정보에 대한 질의 시 수 많은 테이블들간 조인 연산이 많이 발생가능하며, 오히려 조인 연산으로 인해 하나의 인스턴스 테이블에 모든 인스턴스가 저장되어 있는 Sesame 보다 더욱 검색 시간이 증가할 수 있다. 표 3은 클래스 인스턴스 정보를 위한 테이블 구조를 보여준다.

3. XPOS(XPath-based OWL Storage) 모델

이 장에서는 기존의 시스템들에서 계층 구조 저장에 대한 문제점을 해결하고 보다 효율적이고 빠른 OWL 문서 정보 검색을 위한 효율적인 OWL 저장 모델인 XPOS(XPath-based OWL Storage) 모델을 제안하고 OWL 문서를 입력으로 받아 XPOS 모델에 변환하여 저장하기 위한 XPOS 시스템 구조에 대해 기술한다.

3.1 XPOS 모델 정의

XPOS 모델은 클래스/프로퍼티들 간의 계층 관계를 모두 포함하며 계층 구조에 대한 효율적인 검색이 가능하도록 설계되었다. 표 4는 XPOS 모델의 저장 스키마 구조를 나타낸다.

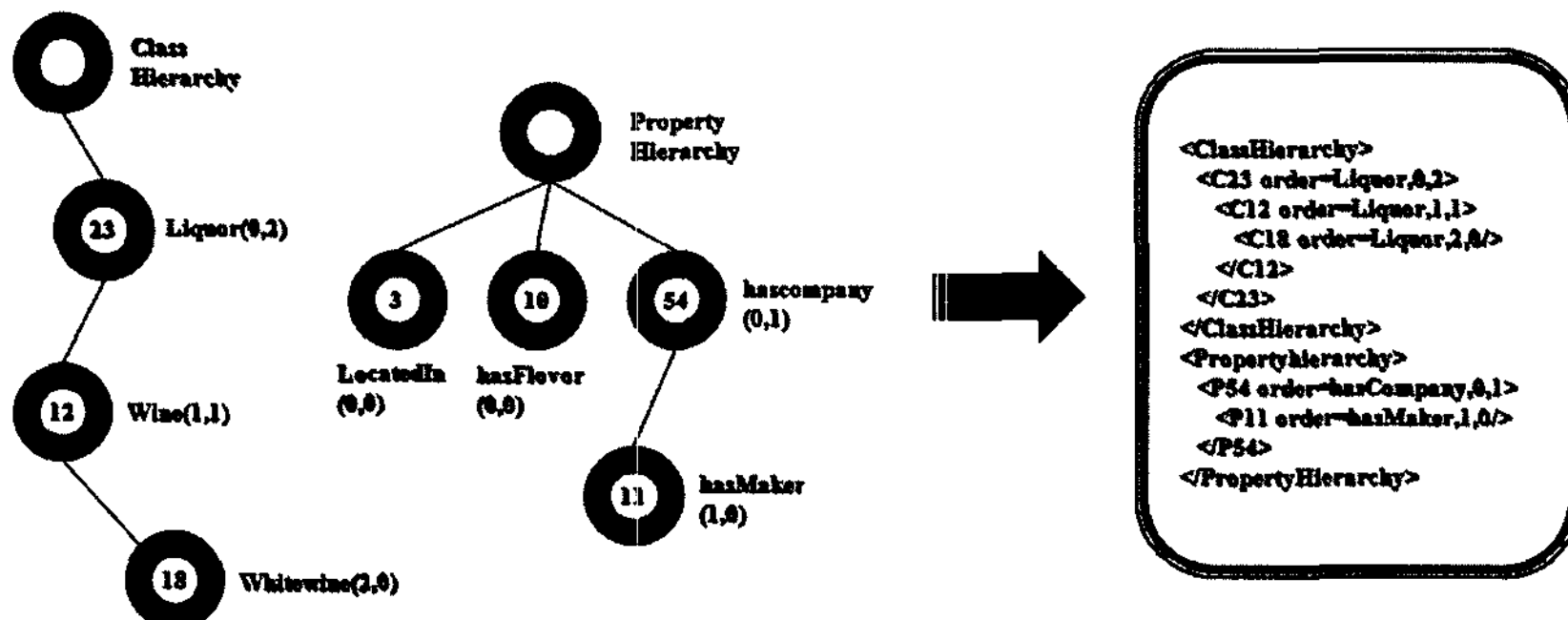


그림 3 계층 정보 저장을 위한 XML 문서

표 3 클래스 인스턴스 정보 위한 테이블

preorder	postorder	uid	preorder	postorder	class uid	value_u	value_s
"Classname" Table			"Propertyname" Table				

표 4 XPOS 모델의 저장 스키마

class_id	class_name	class_path	root_id	inst_id	inst_name	class
Class Table				Instance Table		
prop_id	prop_name	prop_path	root_id	subject	predicate	object
Property Table				Triples Table		

XPOS 모델의 저장 스키마는 크게 4개 테이블, Class, Property, Triple, Instance로 구성되며, 계층 구조 정보를 제외한 기본 구조는 Sesame 구조와 유사하다. 우선 Class 테이블과 Property 테이블은 각 클래스 및 프로퍼티의 개별적인 식별을 위한 id, 그리고 이름 정보를 포함한다. 또한 path 속성과 root_id 속성을 포함하여 클래스/프로퍼티간 계층 구조 관련 정보를 포함한다. path 속성에는 계층 구조 정보가 'Student/Graduate School Student/Ph. D. Student'와 같이 XPath의 형태로 저장된다. Instance 테이블은 각 클래스에 해당하는 인스턴스를 식별하기 위한 고유한 id, 인스턴스의 이름, 그리고 해당하는 클래스의 id 정보를 포함한다. Triples 테이블은 클래스 또는 클래스 인스턴스와 프로퍼티의 관계 정보를 포함하며, subject-predicate-object 트리플 구조로 저장된다. subject 속성과 object 속성에는 class_id 또는 inst_id 값이 저장될 수 있으며, predicate 속성에는 prop_id 값이 저장된다.

입력된 OWL 데이터로부터 각 클래스의 경로 추출을 위해서는 다음과 같은 과정을 거치게 된다. 우선 입력된 OWL 데이터의 스키마 부분을 분석하여 상속 관계를 가지는 클래스와 프로퍼티들을 데이터 그래프화시킨다 [14-16]. 그래프로 표현된 OWL 데이터를 기반으로 단말 클래스/프로퍼티까지 깊이우선 탐색기법을 적용하여 모든 데이터가 단 한 번만 표현 되도록 탐색하여 단말 클래스/프로퍼티까지 전체 경로를 추출한다. 추출된 정보에는 최상위 클래스/프로퍼티부터 단말 클래스/프로퍼티까지의 경로가 포함되며 이 정보는 Class 테이블과 Property 테이블의 Path 속성에 저장된다. 또한 하나의 OWL 문서에 여러 개의 온톨로지가 포함될 경우, 검색 시 또는 온톨로지 내용 변경 시 해당 온톨로지에 대한 빠른 접근 및 검색을 좀 더 용이하도록 하기 위해 각 클래스/프로퍼티에 대한 최상위 클래스/프로퍼티의 ID를 Class 테이블과 Property 테이블에 저장한다. 즉 온톨로지 변경이나 재구성 시, 우선 최상 클래스/프로퍼티를 확인한 후 변경 또는 재구성해야 할 온톨로지의 최

상위 클래스/프로퍼티에 속하는 자식 클래스/프로퍼티만 변경 또는 재구성할 수 있다.

그림 4는 간단한 OWL 문서의 클래스 계층 구조와 Class 테이블의 Path 속성에 저장된 계층 구조 정보를 나타낸다. 그림 4와 같이 각 클래스/프로퍼티의 계층 구조 정보가 저장되는 경우, 기존 시스템과는 달리 반복적인 작업이나 XML 저장소와 같은 다른 저장소에 접근할 필요 없이 클래스/프로퍼티 테이블 내에서 효율적으로 클래스의 계층 정보를 추출해 낼 수 있다. 즉, 특정 클래스/프로퍼티에 대한 계층 정보는 Path 속성 값을 통해 알 수 있으며, 특정 클래스/프로퍼티의 하위 클래스/프로퍼티를 추출할 경우에도 Path 속성만 탐색하여 경로 내에 특정 클래스/프로퍼티 이름을 포함하는 클래스를 모두 추출하면 간단하게 하위 클래스/프로퍼티를 추출해 낼 수 있다. 따라서 입력된 OWL 데이터가 복잡하고 깊이가 깊은 계층 구조인 경우라도 계층 구조를 추출할 때는 Path 속성만 탐색하여 효율적으로 각 클래스의 계층 구조를 추출해 낼 수 있다.

하지만, 최근 테라바이트 또는 페타바이트들과 같이 초대용량 데이터가 데이터베이스에 저장, 관리 되는 추세이므로, 이런 경우 깊이가 깊고 복잡한 계층 구조를 가진 데이터가 존재할 수 있으며, 이를 저장하기 위해서는 path 저장 부분에 대한 공간적인 문제가 발생할 수도 있다. 하지만, class_path와 property_path 컬럼에 저장되는 정보는 단순한 스트링 데이터이며, 일반적으로 초대용량의 데이터라 하더라도 구조가 복잡하기 보다는 인스턴스 값이 많은 경우가 대부분이므로 이러한 문제는 향후 연구로 남겨둔다.

3.2 XPOS 시스템 구조

그림 5는 OWL 데이터를 저장하기 위한 XPOS 시스템 구조이다. OWL 문서가 입력되면 OWL 파서가 문서를 파싱한다. 이 때 OWL 파서는 클래스, 프로퍼티, 인스턴스 정보 및 계층 정보, 제약 조건을 OWL 문서로부터 추출한다. 파싱 결과로 추출된 데이터들을 기반으로 클래스/프로퍼티 간의 계층정보는 계층 정보 추출기에

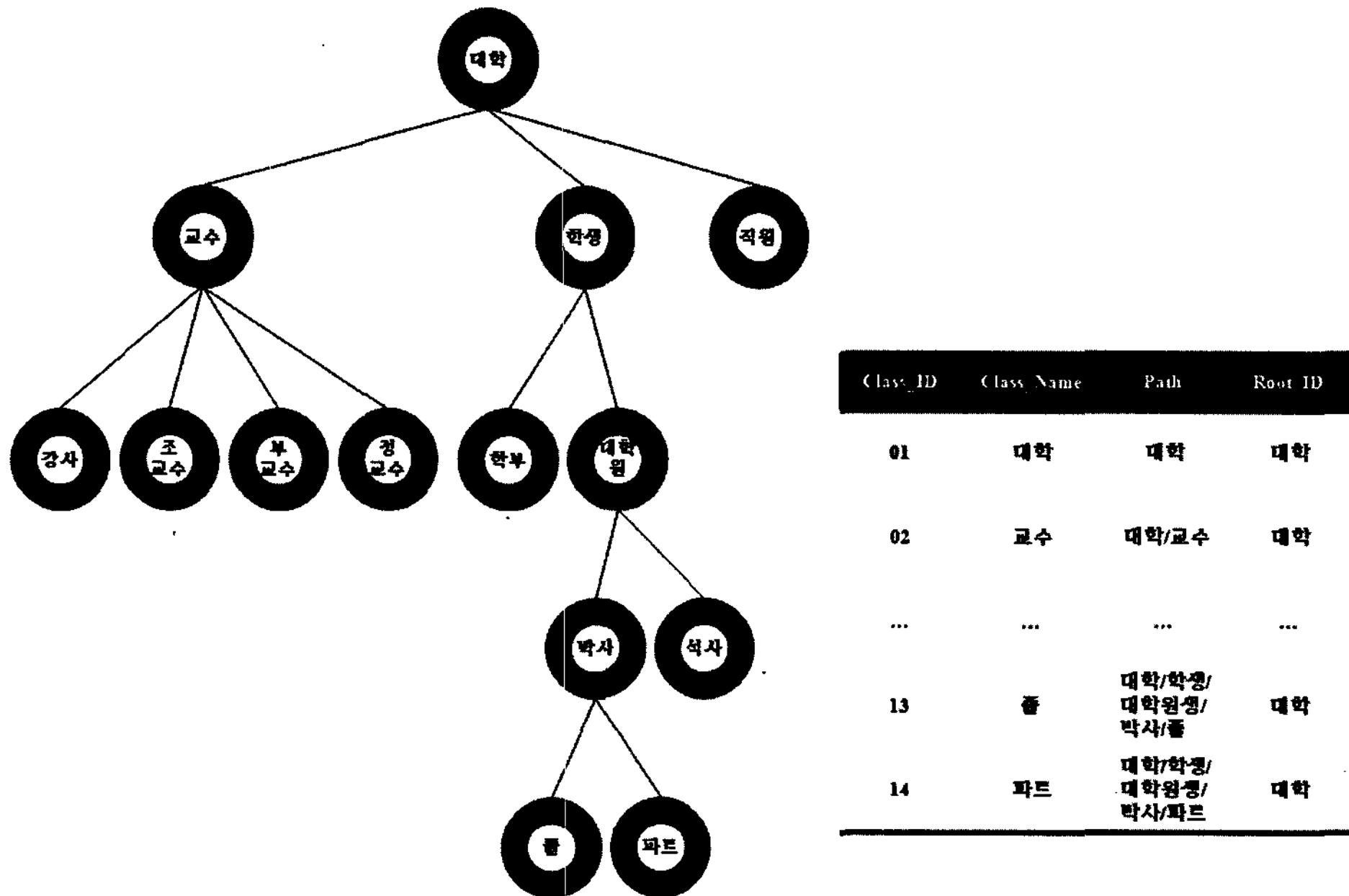


그림 4 클래스에 대한 데이터 그래프와 클래스 테이블의 계층 구조 정보

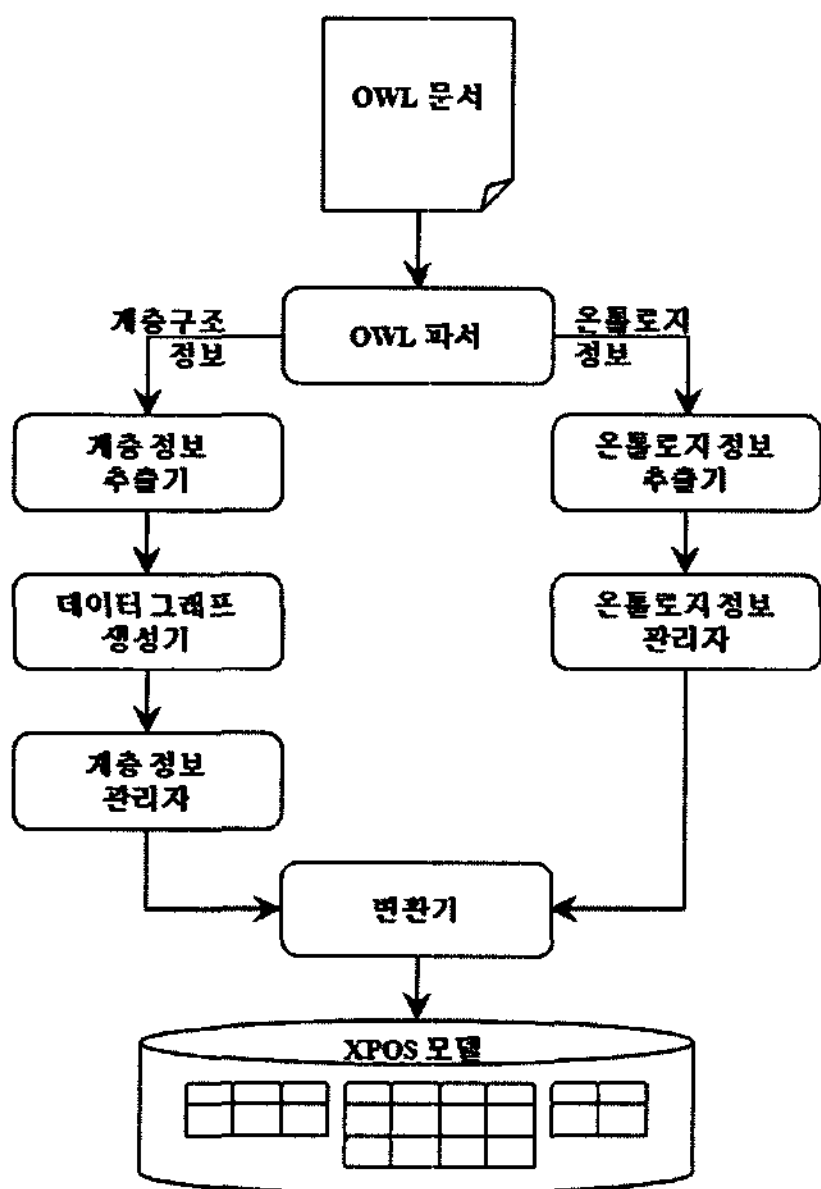


그림 5 XPOS 시스템 구조

의해 분류 추출된다. 추출된 클래스/프로퍼티간 계층 정보는 데이터 그래프 생성기에 의해 데이터 그래프로 변환 생성된다. 데이터 그래프는 그림 5와 같은 형태로 생성되며 계층 정보 관리자는 생성된 데이터 그래프에 기반으로 단말 클래스/프로퍼티까지 깊이 우선 탐색 기법을 적용함으로써 모든 데이터를 단 한 번만 표현되도록

단말 클래스/프로퍼티까지 전체 경로를 추출한다. 클래스/프로퍼티간 계층 정보를 제외한 클래스, 프로퍼티, 인스턴스 정보는 OWL 파서의 파싱 결과를 기반으로 온톨로지 정보 추출기에 의해 분류 추출된다. 온톨로지 정보 관리자는 클래스와 인스턴스 간 관계, 클래스와 프로퍼티간의 관계를 추출하여 정리한다. 최종적으로, 계층 정보 관리자와 온톨로지 정보 관리자에 의해 추출 정리된 계층 구조 정보와 온톨로지 정보는 변환기에 의해 XPOS 모델로 변환 및 저장된다. 세부적인 시스템 동작과정이나 OWL 데이터의 변환 과정은 3.3절에서 예시와 함께 기술한다.

3.3 XPOS 시스템의 저장 과정

이 절에서는 XPOS 시스템의 변환 및 저장 과정을 기술하기 위해 Univ-Bench Artificial data generator (UBA)에 의해 생성된 OWL 데이터의 일부를 사용한다.

표 5는 UBA에 의해 제공되는 University0_0.owl 문서의 일부로서 대학교, 학과, 활동 등의 포괄적인 대학 내용 중 대학교 구성원 데이터만을 변환 과정을 보이기 위한 샘플 예제로 사용한다. 전체 OWL 문서는 복잡하고 양이 방대하므로 표 5에서는 그 일부만을 기술한다. 표 5와 같은 OWL 문서가 입력이 되면 OWL 파서에 의해 파싱이 되며 구문이나 문법 상의 오류를 체크한다. 파싱 시 오류가 발생하면 이 OWL 문서에 대한 변환 처리는 이 단계에서 중단된다. 파싱이 완료되고 구문, 문법상의 오류가 발견되지 않은 경우 OWL 문서는 계층 정보 추출기와 온톨로지 정보 추출기에 의해 분석되

표 5 예제 OWL 문서

<pre><?xml version="1.0" encoding="UTF-8" ?> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:ub="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#"> <owl:Ontology rdf:about=""> <owl:imports rdf:resource="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl" /> </owl:Ontology> <ub:University rdf:about="http://www.University0.edu"> <ub:name>University0</ub:name> </ub:University> <ub:Department rdf:about="http://www.Department0.University0.edu"> <ub:name>Department0</ub:name> <ub:subOrganizationOf> <ub:University rdf:about="http://www.University0.edu" /> </ub:subOrganizationOf> </ub:Department> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor0"> <ub:name>http://www.Department0.University0.edu/FullProfessor0</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course0" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu" /> <ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University653.edu" /> </ub:doctoralDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor2@Department0.University0.edu</ub:emailAddress> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research28</ub:researchInterest> </ub:FullProfessor> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor3"> <ub:name>http://www.Department0.University0.edu/FullProfessor3</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course4" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course5" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse4" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse5" /> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University74.edu" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University336.edu" /> </ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University112.edu" /> </ub:doctoralDegreeFrom> </ub:FullProfessor> </pre>	<pre>niversity0.edu/GraduateCourse0" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse1" /> </ub:teacherOf> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University84.edu" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University875.edu" /> </ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University241.edu" /> </ub:doctoralDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor0@Department0.University0.edu</ub:emailAddress> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research20</ub:researchInterest> </ub:FullProfessor> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor1"> <ub:name>http://www.Department0.University0.edu/FullProfessor1</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course1" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse2" /> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University260.edu" /> </ub:undergraduateDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor3@Department0.University0.edu</ub:emailAddress> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research12</ub:researchInterest> </ub:FullProfessor> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor4"> <ub:name>http://www.Department0.University0.edu/FullProfessor4</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course6" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse6" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse7" /> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University608.edu" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University737.edu" /> </ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University143.edu" /> </ub:doctoralDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor4@Department0.University0.edu</ub:emailAddress> </pre>	<pre>" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University503.edu" /> </ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University882.edu" /> </ub:doctoralDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor1@Department0.University0.edu</ub:emailAddress> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research12</ub:researchInterest> </ub:FullProfessor> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor2"> <ub:name>http://www.Department0.University0.edu/FullProfessor2</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course2" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course3" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse3" /> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University862.edu" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University125.edu" /> </ub:mastersDegreeFrom> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research7</ub:researchInterest> </ub:FullProfessor> <ub:FullProfessor rdf:about="http://www.Department0.University0.edu/FullProfessor5"> <ub:name>http://www.Department0.University0.edu/FullProfessor5</ub:name> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/Course7" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse8" /> <ub:teacherOf rdf:resource="http://www.Department0.University0.edu/GraduateCourse9" /> <ub:undergraduateDegreeFrom> <ub:University rdf:about="http://www.University451.edu" /> </ub:undergraduateDegreeFrom> <ub:mastersDegreeFrom> <ub:University rdf:about="http://www.University718.edu" /> </ub:mastersDegreeFrom> <ub:doctoralDegreeFrom> <ub:University rdf:about="http://www.University358.edu" /> </ub:doctoralDegreeFrom> <ub:worksFor rdf:resource="http://www.Department0.University0.edu" /> <ub:emailAddress>FullProfessor5@Department0.University0.edu</ub:emailAddress> <ub:telephone>xxx-xxx-xxxx</ub:telephone> <ub:researchInterest>Research18</ub:researchInterest> </ub:FullProfessor> </pre>
--	--	--

며, 분석 과정을 통해 OWL 문서를 클래스/프로퍼티간 계층 정보를 포함하는 부분과 클래스/프로퍼티/인스턴스 및 트리플 정보를 포함하는 부분으로 분류한다. OWL 문서의 분석 및 분류가 완료되면 분류된 클래스/프로퍼티간 계층 정보는 데이터 그래프 생성기에 의해 데이터 그래프화 된다. 그림 6은 클래스와 프로퍼티 계층 구조에 대한 데이터 그래프이다. 데이터 그래프가 완성되면 계층 정보 관리자에 의해 최상위 노드를 시작으로 단말 노드까지 깊이 우선 탐색 기법을 이용해 탐색을 하며, 단말 노드에 이른 경우 해당 노드의 경로 정보(path)를 생성한다. 이때 탐색한 중간 노드들에 대한 경로 정보도 함께 생성한다. 단, 단말 노드의 형제 노드 탐색 시에는 중간 노드들에 대한 경로 정보는 최초 1회만 생성함으로써 중복 경로 생성을 방지한다. 표 6은 클래스 계층 구조 그래프와 프로퍼티 계층 구조 그래프의 탐색 순서 및 탐색 경로, 그리고 탐색 시 생성되는 경로정보를 나타낸다.

표 6과 같이 깊이 우선 탐색 순서에 따라 모든 노드를 탐색한 후 각 노드들의 path가 모두 생성되면 계층 구조는 Class 테이블과 Property 테이블의 path 컬럼에 저장되며, 클래스와 프로퍼티 이름, 인스턴스 값, subject-predicate-object의 트리플 구조는 Class 테이블, Property 테이블, Instance 테이블, Triples 테이블에 각각 저장된다. Class/Property/Instance 테이블의 id는 순차적으로 부여되어 저장되며, 계층 정보 관리자에 의해 생성된 계층 구조는 Class/Property 테이블의 path 컬럼에 XPath 형태로 저장된다. 또한 Class/Property 테이블의 root_id 컬럼에는 변경 및 재구성의 편의를 위하여 각 클래스나 프로퍼티에 대한 데이터 그래프에서의 최상위 노드 정보를 저장한다. Instance 테이블에는 인스턴스의 이름과 인스턴스의 해당 클래스 정보 또한 저장되며, Triples 테이블에는 클래스-프로퍼티-인스턴스/클래스의 형태로 각각 정보가 URL 형태로 저장된다. 그림 7은 표 5의 입력 OWL 문서가 XPOS 시스템에 의

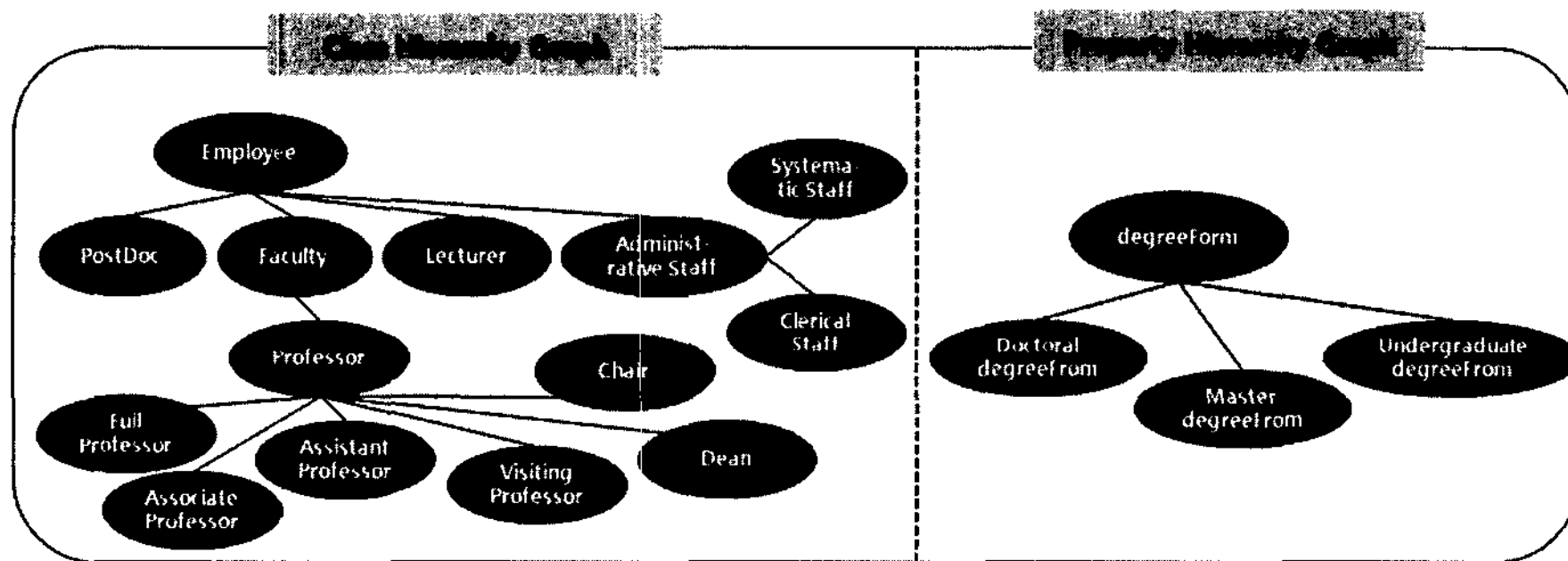


그림 6 클래스/프로퍼티들의 데이터 그래프

표 6 데이터 그래프 탐색 순서 및 경로 정보 생성

순서	탐색 경로	생성 path
1	[Class] Employee->PostDoc	Employee/PostDoc
2	[Class] Employee->Faculty->Professor->Full Professor	Employee/Faculty Employee/Faculty/Professor Employee/Faculty/Professor/Full Professor
3	[Class] Employee->Faculty->Professor->Associate Professor	Employee/Faculty/Professor/Associate Professor
4	[Class] Employee->Faculty->Professor->Assistant Professor	Employee/Faculty/Professor/Assistant Professor
5	[Class] Employee->Faculty->Professor->Visiting Professor	Employee/Faculty/Professor/Visiting Professor
6	[Class] Employee->Faculty->Professor->Dean	Employee/Faculty/Professor/Dean
7	[Class] Employee->Faculty->Professor->Chair	Employee/Faculty/Professor/Chair
8	[Class] Employee->Lecturer	Employee/Lecturer
9	[Class] Employee->Administrative Staff->Systematic Staff	Employee/Administrative Staff Employee/Administrative Staff/Systematic Staff
10	[Class] Employee->Administrative Staff->Clerical Staff	Employee/Administrative Staff/Clerical Staff
11	[Property] degreeFrom->Doctoral degreeFrom	degreeFrom/Doctoral degreeFrom
12	[Property] degreeFrom->Master degreeFrom	degreeFrom/Master degreeFrom
13	[Property] degreeFrom->Undergraduate degreeFrom	degreeFrom/Undergraduate degreeFrom

Class Table				Property Table			
class_id	class_name	class_path	root_id	prop_id	prop_name	prop_path	root_id
c01	Employee	Employee	Employee	p01	degreeFrom	degreeFrom	degreeFrom
c02	PostDoc	Employee/PostDoc	Employee	p02	Doctoral degreeFrom	degreeFrom/Doctoral degreeFrom	degreeFrom
c03	Faculty	Employee/Faculty	Employee	p03	Master degreeFrom	degreeFrom/Master degreeFrom	degreeFrom
c04	Professor	Employee/Faculty/Professor	Employee	p04	Undergraduate degreeFrom	degreeFrom/Undergraduate degreeFrom	degreeFrom
c05	Full Professor	Employee/Faculty/Professor/Full Professor	Employee				
c06	Associate Professor	Employee/Faculty/Professor/Associate Professor	Employee				
c07	Assistant Professor	Employee/Faculty/Professor/Assistant Professor	Employee				
c08	Visiting Professor	Employee/Faculty/Professor/Visiting Professor	Employee				
c09	Dean	Employee/Faculty/Professor/Dean	Employee				
c10	Chair	Employee/Faculty/Professor/Chair	Employee				
c11	Lecturer	Employee/Lecturer	Employee				
c12	Administrative Staff	Employee/Administrative Staff	Employee				
c13	Systematic Staff	Employee/Administrative Staff/Systematic Staff	Employee				
c14	Clerical Staff	Employee/Administrative Staff/Clerical Staff	Employee				

Instance Table		
inst_id	inst_name	class
i01	http://www.Department0.University0.edu/AssistantProfessor0	Assistant Professor
i02	http://www.Department0.University0.edu/FullProfessor2	Full Professor
i03	http://www.Department0.University0.edu/FullProfessor8	Full Professor
i04	http://www.Department0.University0.edu/AssociateProfessor5	Associate Professor
i05	http://www.Department0.University0.edu/AssociateProfessor7	Associate Professor
i06	http://www.Department0.University0.edu/AssistantProfessor9	Assistant Professor
i07	http://www.Department0.University0.edu/AssistantProfessor2	Assistant Professor
i08	http://www.Department0.University0.edu/UndergraduateStudent324	Undergraduate degreeFrom
i09	http://www.Department0.University0.edu/UndergraduateStudent229	Undergraduate degreeFrom
i10	http://www.Department0.University0.edu/Lecturer1	Lecturer
----	-----	-----

Triples Table		
subject	property	object
http://www.Department5.University0.edu/AssistantProfessor4	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#doctoralDegreeFrom	http://www.University932.edu
http://www.Department5.University0.edu/AssistantProfessor4	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#mastersDegreeFrom	http://www.University264.edu
http://www.Department5.University0.edu/AssistantProfessor5	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#undergraduateDegreeFrom	http://www.University939.edu
http://www.Department5.University0.edu/AssistantProfessor6	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#mastersDegreeFrom	http://www.University965.edu
http://www.Department4.University0.edu/GraduateStudent126	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#undergraduateDegreeFrom	http://www.University307.edu
http://www.Department4.University0.edu/GraduateStudent74	http://www.lehigh.edu/~zhp2/2004/0401/uriv-bench.owl#undergraduateDegreeFrom	http://www.University33.edu
----	-----	-----

그림 7 XPOS 모델 저장 예시

해 XPOS 저장 모델에 저장된 모습이다.

4. 성능 평가

이 장에서는 이 논문에서 제안하고 있는 XPOS 시스템과 Sesame 및 XML 저장소 기반 저장 모델간 질의 처리 성능에 대한 실험 및 평가 결과에 대하여 기술한다.

4.1 실험 환경 및 데이터

실험에서 사용된 시스템의 사양은 Pentium Dual CPU 2.66GHz이며, 메모리는 1GB이다. DBMS는 Oracle 9i를 사용하였으며 구현 언어로는 Java를 이용하였다.

실험에 사용한 데이터는 UBA에 의해 생성된 데이터셋이다. UBA[17,18]는 Lehigh 대학에서 개발한 OWL 온톨로지 생성 도구이며 다양한 크기의 온톨로지를 생성할 수 있다. 생성된 온톨로지 데이터는 대학교와 학과, 활동 등에 대한 내용으로 구성되며, 총 43개의 클래스와 32개의 프로퍼티를 포함한다. 이 논문에서는 UBA를 이용하여 1MB, 5MB, 10MB, 세 가지 OWL 온톨로지를 생성하였으며, 각각의 OWL 온톨로지를 Sesame, XML 저장소 기반 저장 모델, XPOS 시스템을 통해 RDBMS에 변환 및 저장한 후, 각 시스템 구조에 적절한 질의를 실행하여 처리 시간을 비교 평가한다. 그림 8은 UBA 온톨로지 데이터의 클래스/프로퍼티 계층 구조를 나타낸다.

질의 처리 시간에 대한 비교 평가를 위해 실험에 사용된 질의는 총 6가지이다. 우선 Q1은 특정 클래스에 대한 모든 하위 클래스를 검색하는 질의이며, Q2는 특정 클래스에 대한 모든 하위 클래스를 검색한 후 각 클래스

들에 해당하는 인스턴스 전체를 검색하는 질의이다. Q3는 특정 프로퍼티에 대한 모든 하위 프로퍼티를 검색하는 질의이며, Q4는 특정 프로퍼티에 대한 모든 하위 프로퍼티를 검색한 후 각 프로퍼티와 관련이 있는 object를 검색하는 질의이다. Q5는 특정 프로퍼티에 대한 모든 하위 프로퍼티를 검색한 후 각 프로퍼티와 관련이 있는 subject를 검색하는 질의이며, Q6은 특정 프로퍼티에 대한 모든 하위 프로퍼티를 검색한 후 각 프로퍼티와 관련이 있는 object를 검색하고, 그 object에 해당하는 인스턴스가 포함되는 클래스를 검색하는 질의이다.

성능 평가에 사용되는 모든 질의는 클래스 또는 프로퍼티간 계층 구조에 대한 검색을 포함하고 있으며, 실험에서도 계층 구조의 저장 기법 별 처리 성능 차이에 대한 비교 평가에 중점을 둔다. 즉, 성능 평가에서는 많은 클래스와 프로퍼티들이 계층 구조를 지니고 있을 때 클래스/프로퍼티간 관계를 추출해 내는 성능 비교에 초점을 둔다. 표 7은 6개 질의에 대한 SPARQL 표현과 각 질의의 의미에 대해 기술한다.

4.2 실험 및 평가 결과

그림 9는 Q1에 대한 각 모델의 질의 처리 결과이다. OWL 파일의 크기가 증가해도 전체 클래스의 개수는 변하지 않으므로, 특정 클래스의 하위클래스에 대한 결과는 변하지 않는다. 따라서 파일의 크기가 증가해도 질의 처리 시간은 모델 별로 거의 일정하다. Sesame는 subClassOf 테이블에서 'Employee' 클래스의 하위 클래스를 찾기 위해 하위 클래스 검색을 반복 실행해야 하며, XML 저장소 기반 저장 모델은 Employee 클래스

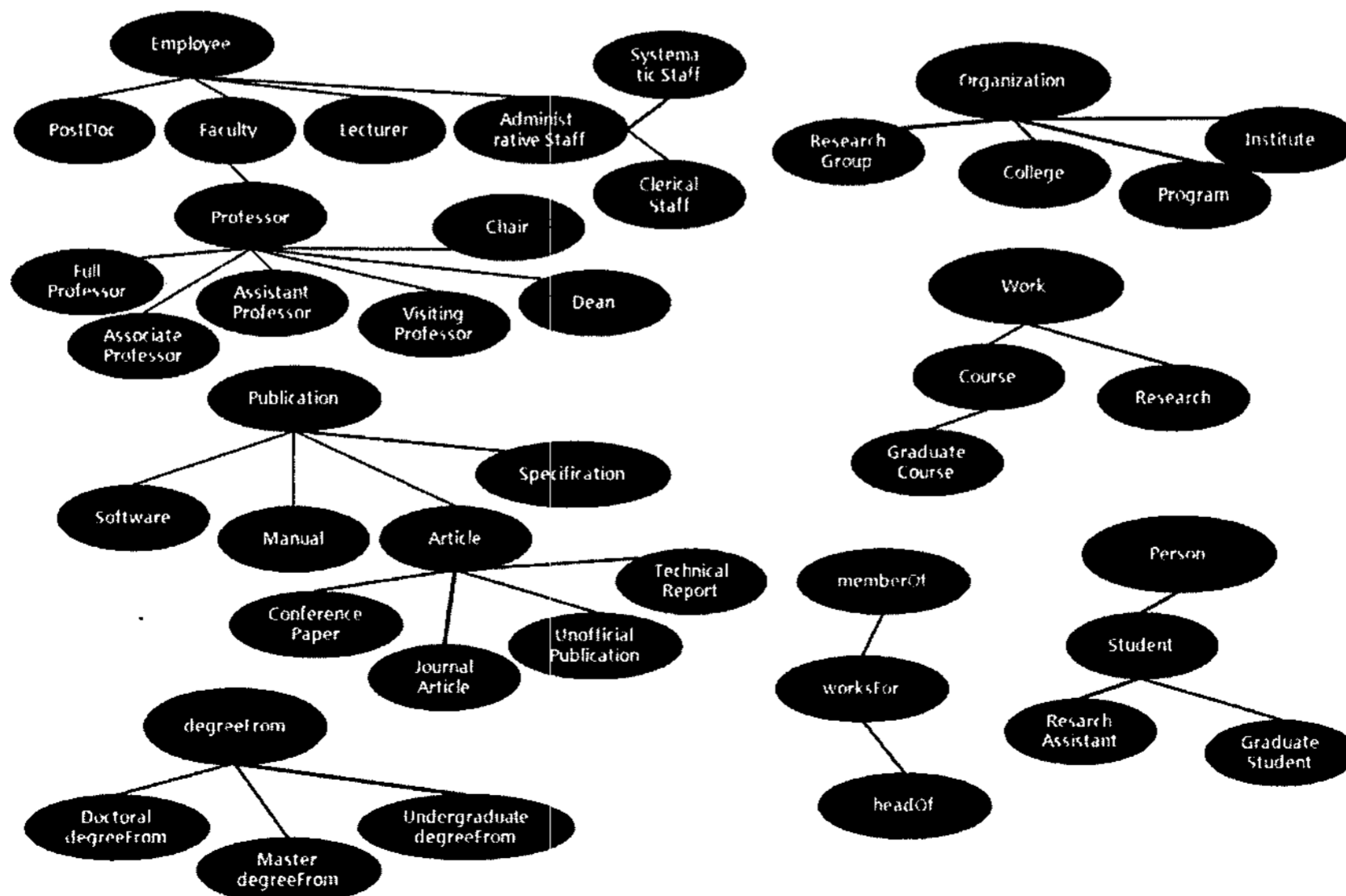


그림 8 UBA 온톨로지 데이터 계층 구조

표 7 실험에 사용된 질의에 대한 SPARQL 표현

각 질의에 대한 SPARQL 표현	
Q1	<pre> select ?class where { {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?x. ?x Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } union all {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } union all {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } } </pre> <p><'Employee' 클래스의 모든 하위 클래스를 검색하는 질의 ></p>
Q2	<pre> select ?instance where { ?instance rdf:type ?class { {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?x. ?x Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } union all {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } union all {?class Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Employee } } </pre> <p><'Employee' 클래스의 모든 하위 클래스를 검색하여 각 클래스에 해당하는 인스턴스 값을 검색하는 질의></p>
Q3	<pre> select ?property where { {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } union all {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } } </pre> <p><'memberOf' 프로퍼티의 모든 하위 프로퍼티를 검색하는 질의 ></p>
Q4	<pre> select ?object where { ?x ?property ?object { {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } union all {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } } </pre> <p><'memberOf' 프로퍼티의 모든 하위 프로퍼티를 검색하여 각 프로퍼티에 해당하는 Object 값을 검색하는 질의 ></p>
Q5	<pre> select ?subject where { ?subject ?property ?x { {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } union all {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } } </pre> <p><'memberOf' 프로퍼티의 모든 하위 프로퍼티를 검색하여 각 프로퍼티에 해당하는 Subject 값을 검색하는 질의 ></p>
Q6	<pre> select ?class where { ?object rdf:type ?class. ?x ?property ?object { {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf ?y. ?y Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } union all {?property Uv::http://www.w3.org/2000/01/rdf-schema#subPropertyOf Uv::http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf } } </pre> <p><'memberOf' 프로퍼티의 모든 하위 프로퍼티를 검색하여 각 프로퍼티에 해당하는 object 값을 검색 한 후 object의 인스턴스 값이 포함되는 클래스를 검색하는 질의 ></p>

에 대해 XML 저장소에 접속하여 최상위 클래스를 확인한 후 계층 정보를 토대로 Employee 테이블 검색을 통해 하위 클래스 모두를 검색한다. 하지만 XPOS 모델의 경우 클래스 테이블 내에서 path 속성의 값만을 탐색하여 하위 클래스를 검색할 수 있다. 따라서 Q1의 경우 하나의 테이블만을 검색하여 질의를 수행하는 XPOS 모델이 가장 좋은 성능을 보인다.

그림 10은 Q2에 대한 각 모델의 질의 처리 결과이다. Q2는 파일의 크기가 커짐에 따라 인스턴스 개수가 증가함으로써 질의 처리 시간 또한 증가한다. Sesame와 XPOS 모델의 경우는 Q1의 결과를 통해 instanceOf/Instance 테이블을 검색하여 해당하는 인스턴스들을 검색한다. 인스턴스 검색은 인스턴스 정보를 저장하는 하나의 테이블 검색을 통해 결과를 얻지만, 하위 클래스를 검색하는데 걸리는 시간의 차이로 인해 XPOS 모델이 Sesame에 비해 Q2 수행 결과 또한 우수한 성능을 보인다. 반면, XML 저장소 기반 저장 모델은 Q1의 결과를 통해 클래스/프로퍼티 별로 나누어 저장되어 있는 모든 인스턴스들을 조인 연산을 통해 검색하므로, 다른 두 모델에 비해 높은 질의 처리 비용을 요구한다.

그림 11은 Q3에 대한 각 모델의 질의 처리 결과이다. 파일의 크기가 증가하더라도 프로퍼티 개수는 변하지 않기 때문에 Q3 수행 시간은 파일 크기가 커지더라도 거의 일정한 값을 유지한다. Sesame는 subPropertyOf 테이블에서 'memberOf' 프로퍼티의 하위 프로퍼티를 검색하기 위해 하위 프로퍼티 검색을 반복 실행해야 한다. XML 저장소 기반 저장 모델은 'memberOf' 클래스에 대해 XML 저장소에 접속하여 최상위 클래스를 확인한 후 계층 정보를 토대로 memberOf 테이블 검색을 통해 하위 프로퍼티 모두를 검색한다.

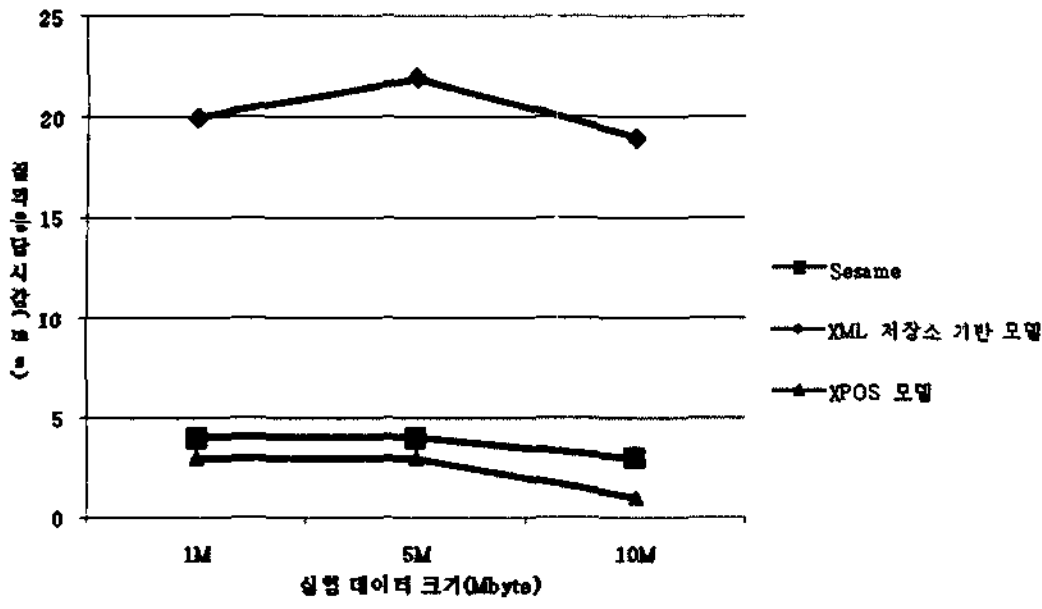


그림 9 Q1에 대한 질의 수행 결과

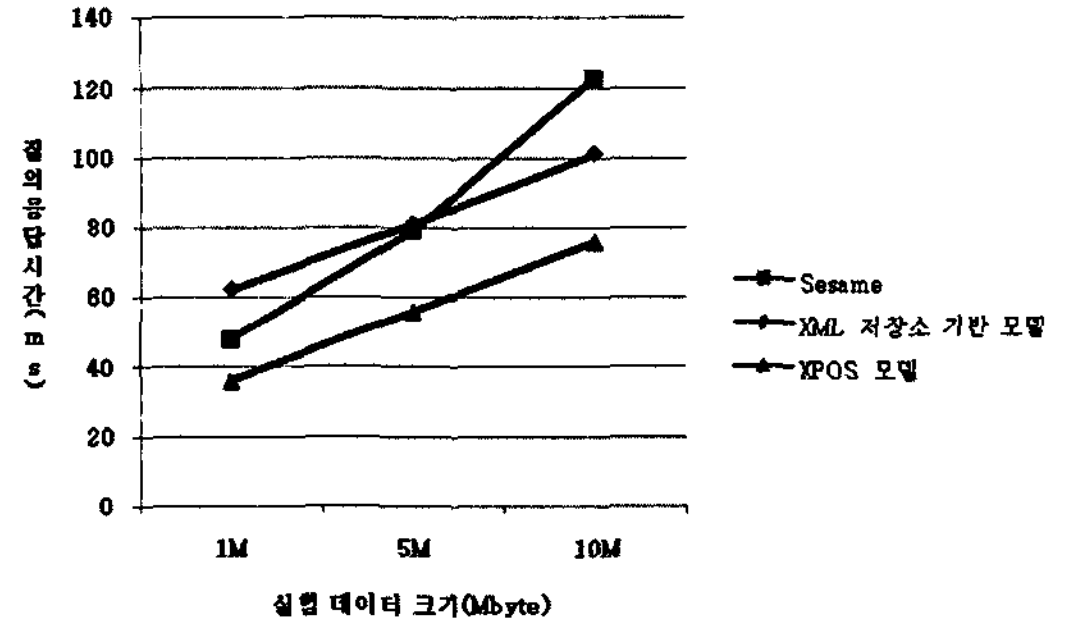


그림 12 Q4에 대한 질의 수행 결과

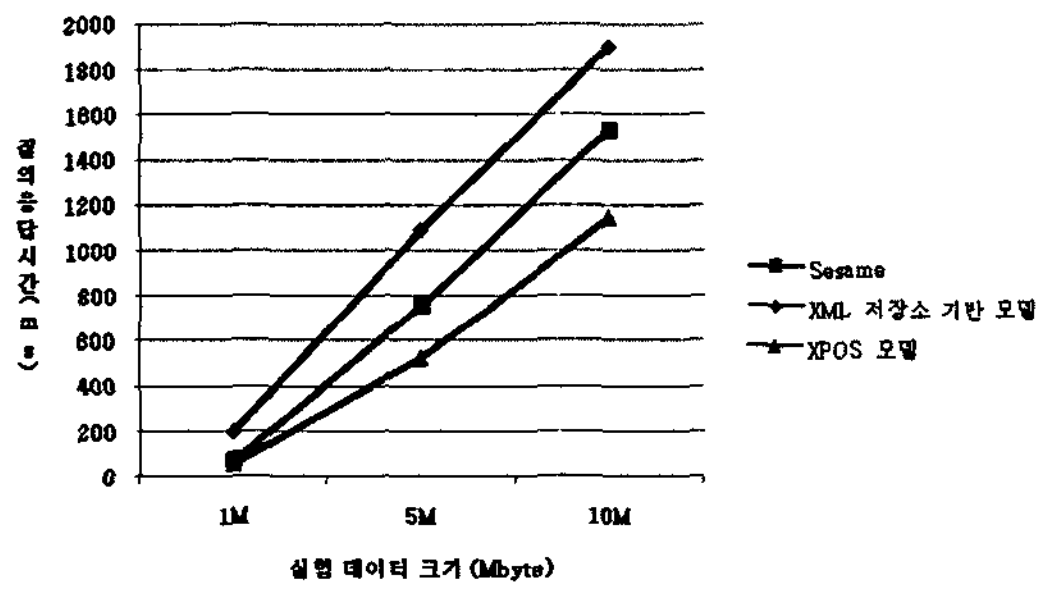


그림 10 Q2에 대한 질의 수행 결과

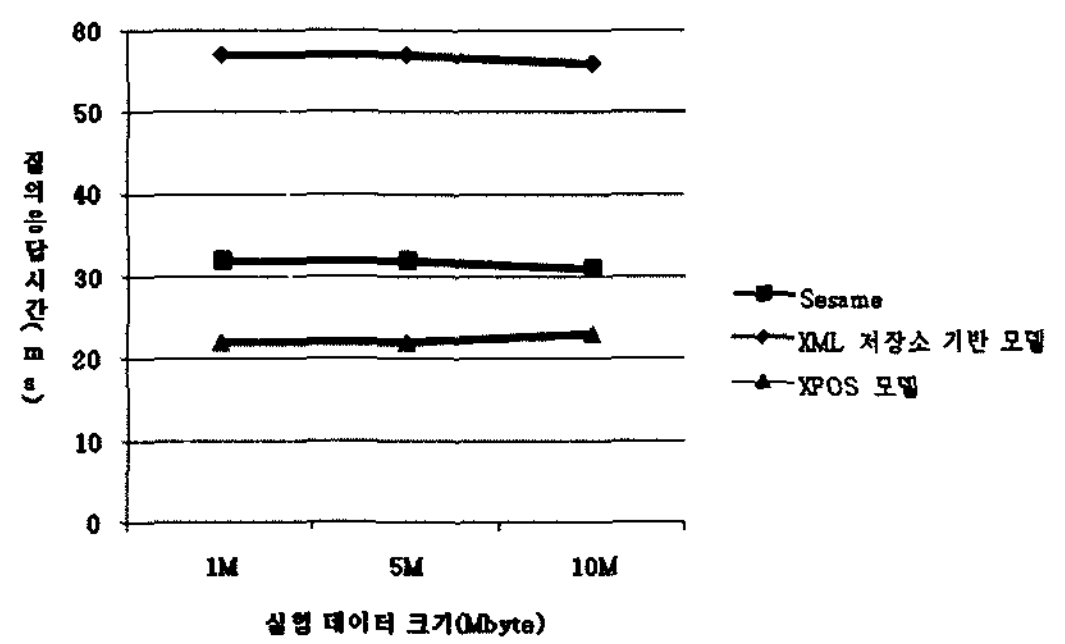


그림 13 Q5에 대한 질의 수행 결과

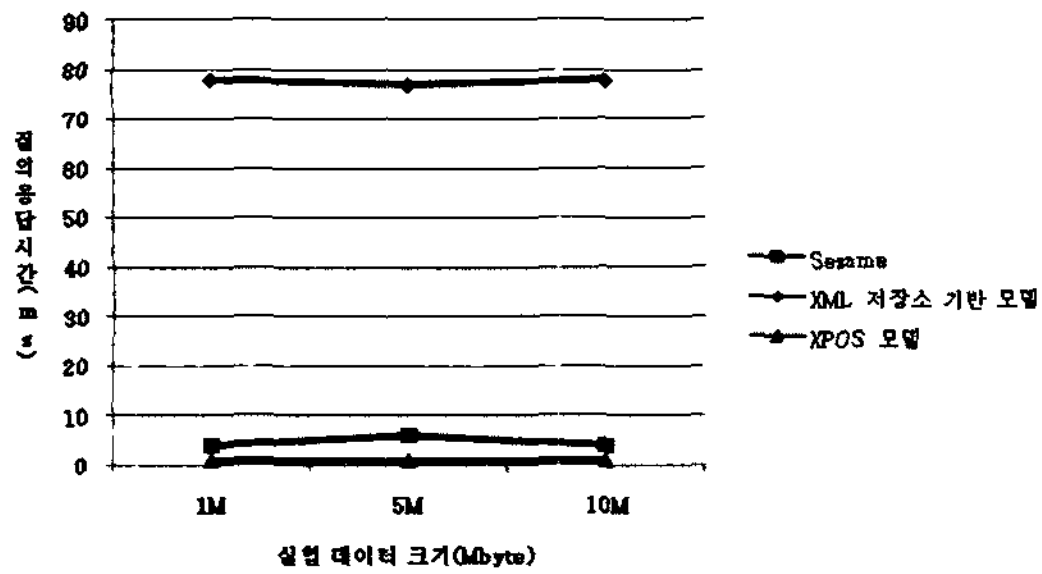


그림 11 Q3에 대한 질의 수행 결과

XML 저장소 기반 저장 모델의 프로퍼티 테이블에는 인스턴스 정보가 함께 저장되어 있기 때문에 프로퍼티의 개수가 적을 경우에도 관련 인스턴스의 개수만큼 테이블을 검색해야 한다. 그러나 XPOS 모델 경우, 프로퍼티 테이블 내에서 path 속성 값을 탐색하여 하위 프로퍼티를 검색할 수 있다. 따라서 Q3 경우 Q1과 마찬가지로 하나의 테이블만을 검색하여 질의를 수행하는 XPOS 모델이 가장 좋은 성능을 보인다.

그림 12와 그림 13은 Q4와 Q5에 대한 각 모델의 질의 처리 실험 결과를 보여준다. 파일 크기가 증가함에 따라 인스턴스 개수가 증가하고 이로 인해 트리플 구조의 object가 증가하게 된다. 따라서 Q4는 파일의 크기가 증가함에 따라 수행 시간이 증가한다. Q4와 Q5는 같은 형태의 질의로써 Q3의 결과를 토대로 검색된 프

로퍼티를 predicate로 지니는 object와 subject를 검색하는 질의이다. Sesame와 XPOS 모델은 Q3의 결과를 통해 Triples 테이블을 검색하여 object와 subject를 얻는다. 프로퍼티의 하위 프로퍼티에 대한 검색 수행 시간의 차이로 인해 Q4와 Q5의 질의 수행 결과도 XPOS 모델이 Sesame에 비해 나은 성능을 보인다. XML 저장소 기반 저장 모델은 Q3의 결과를 토대로 동일 테이블 내에서 object와 subject를 검색하므로 subject와 object에 대한 검색 시 짧은 처리 시간을 보이지만, 프로퍼티의 하위 프로퍼티에 대한 검색 수행 보다 높은 처리 비용을 요구한다. 따라서 XML 저장소 기반 저장 모델은 Q4에 대해서 데이터 크기가 작은 경우에는 Sesame 보다는 좋지 않은 성능을 보이지만 데이터 크기가 증가할수록 Sesame보다 나은 성능을 보인다. 하지만 XML 저장소 기반 저장 모델은 논문에서 제안한 모델인 XPOS 모델보다는 비효율적임을 알 수 있다.

그림 14는 Q6에 대한 각 모델의 질의 처리 결과이다. 파일의 크기가 증가함에 따라 인스턴스의 개수가 증가하고 이로 인해 triple 구조의 object가 증가하게 된다. 따라서 Q6은 파일의 크기가 증가함에 따라 수행 시간이 증가한다. Sesame와 XPOS 모델은 Q4의 결과를 토대로 instanceOf/Instance 테이블에서 해당 인스턴스의 클래스를 검색하게 된다. 프로퍼티의 하위 프로퍼티에 대한 검색 수행 시간의 차이로 인해 Q6의 질의 수행 결과도

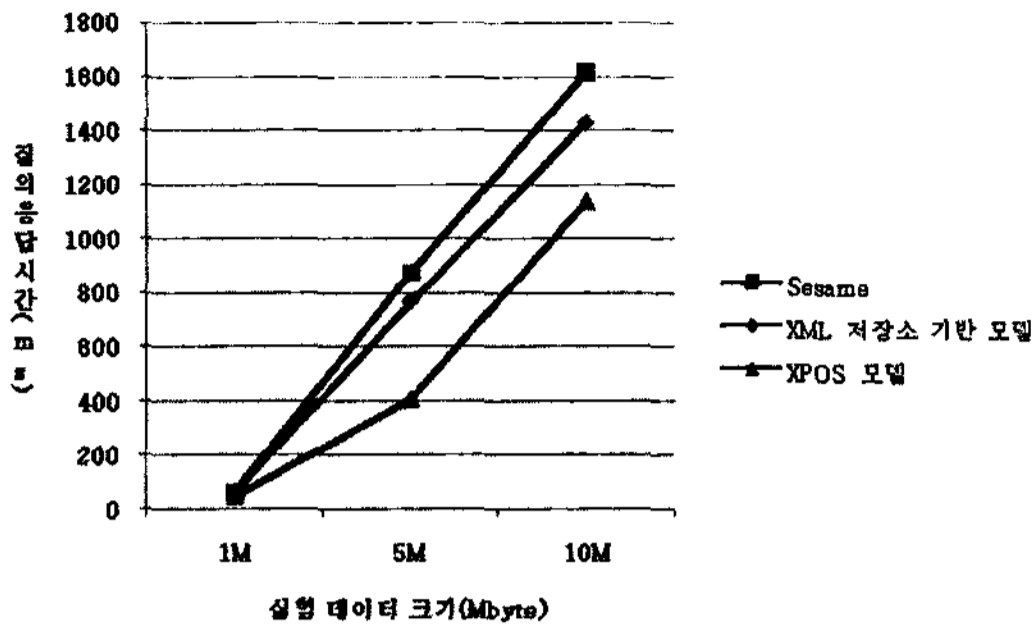


그림 14 Q6에 대한 질의 수행 결과

XPOS 모델이 Sesame에 비해 좋은 성능을 보인다. XML 저장소 기반 저장 모델은 Q4의 결과를 토대로 동일 테이블 내에서 인스턴스의 해당 클래스를 검색하므로 해당 클래스에 대한 검색 수행 시간은 짧지만, 프로퍼티의 하위 프로퍼티에 대한 검색 수행 시간이 오래 걸린다. 따라서 XML 저장소 기반 저장 모델은 Q6에 대해서 Q4와 마찬가지로 Sesame 보다는 좋은 성능을 보이나, XPOS 모델보다는 좋지 않은 성능을 보인다.

앞선 실험 결과, 즉 Q1~Q6의 질의 처리 시간에 대한 실험 결과를 종합하면, 클래스 또는 프로퍼티간 계층 관계 정보를 검색하는데 있어 XPOS 모델의 성능이 Sesame나 XML 저장소 기반 저장 모델에 비해 우수하다는 것을 알 수 있다. Sesame는 앞에서 언급한 바와 같이 계층 관계 정보를 추출해 내기 위해 subClassOf 테이블 또는 subPropertyOf 테이블에서 상위 클래스와 하위 클래스를 검색하는 작업을 반복해야 한다. XML 저장소 기반 저장 모델은 계층 관계 정보 추출을 위해 항상 XML 저장소에 접근해서 XPath 질의를 수행한 후 다시 결과를 토대로 RDBMS에 접근하여 정보를 검색해야 한다. 이에 반해, XPOS 모델은 클래스 또는 프로퍼티간 계층 관련 정보를 추출하기 위해 단지 Class 테이블 또는 Property 테이블 내의 path 속성의 값만을 검색한다. 따라서 클래스 또는 프로퍼티간 계층 정보와 관련이 있는 질의 수행 시에는 항상 XPOS 모델이 다른 두 모델에 비해 우수한 성능을 보인다.

5. 결론 및 향후 연구

생성 및 유통되는 웹 정보의 양이 기하급수적으로 증가함에 따라 사용자가 원하는 정확한 정보 추출이 중요한 이슈로 대두되고 있다. 이러한 배경 속에서 정보에 의미를 부여하고 형식적으로 정의하는 시맨틱 웹이 등장하게 되었으며 RDF, OWL과 같은 온톨로지 기술 언어 개발 및 사용이 활발하게 진행되고 있다. 온톨로지 언어에는 클래스/프로퍼티들간 계층적인 구조 정보가 존재하며 이에 대한 효율적인 저장 관리 모델 개발이 요

구된다. 즉 시맨틱 웹에서 정확한 정보 추출을 위해서는 구조 정보를 고려한 효율적인 OWL 데이터의 저장 기법이 필요하다.

이 논문에서는 OWL 데이터의 효과적이고 정확한 정보 추출을 위해 계층적 구조를 고려한 효율적인 OWL 저장 모델인 XPOS 모델과 OWL 데이터를 변환하여 XPOS 모델로 저장하기 위한 XPOS 시스템 구조에 대하여 기술하였다. XPOS 시스템은 입력된 OWL 데이터의 스키마 부분을 분석하여 상속 관계를 가지는 클래스와 프로퍼티들을 Edge-Labeled Graph에 데이터 그래프화 시킨다. 또한 그래프로 표현된 OWL 데이터를 기반으로 단말(Leaf) 클래스/프로퍼티까지 깊이 우선 탐색 기법을 적용하여 전체 경로를 추출한다. 추출된 계층 정보는 XPOS 모델의 해당 테이블에 XPath 형태로 저장된다.

따라서 이 논문에서 제안한 XPOS 모델 구조로 OWL 데이터를 저장함에 따라 Sesame의 계층 구조 추출을 위한 반복 작업으로 인한 비효율성 및 XML 저장소 기반 저장 모델처럼 이중의 데이터베이스를 사용함에 따라 발생하는 성능 저하의 문제를 극복할 수 있다. 또한 XPOS 모델은 4장에서 기술한 것처럼 기존의 시스템들에 비해 효율적인 질의 처리 성능을 보이며 직관적이고 빠른 계층 정보 추출을 가능하게 한다.

향후 연구로는 OWL 문서의 XPOS 모델로의 저장 효율, 즉 변환저장 시간, 로딩 시간 등이 비교 평가 되어야 할 것이다. 또한 저장 효율과 계층 구조 질의 처리 속도간 상관 관계에 대한 연구 또한 필요할 것이다. 저장 효율이 빠른 경우 저장 시 계층 구조에 대한 고려가 잘 이루어지지 않아 계층 구조 질의에 대한 처리 속도는 느린 경우가 많으며, 저장 시 계층 구조를 상세히 고려하여 저장하는 경우 계층 구조 질의 처리 속도는 빠르고 정확하나 저장 효율은 느릴 수 있다. 따라서 두 가지 모두에 대한 상관 관계를 고려하여 두 가지 효율을 모두 반영하기 위한 연구가 진행되어야 한다.

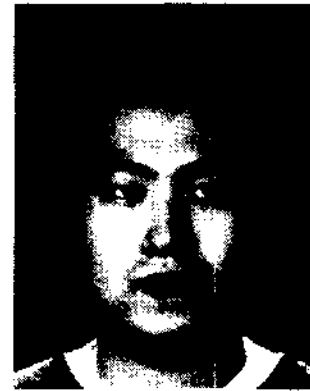
참고 문헌

- [1] Tim Berners-Lee, James Hendler, and Ora Lassila, "The Semantic Web," Scientific American, May 2001.
- [2] Stefan Decker, Sergey Melnik, Frank van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks, "The Semantic Web: The Roles of XML and RDF," IEEE Internet Computing, Vol. 15, no. 3, pp. 63-74, Oct. 2000.
- [3] Ismail Fahmi, Junte Zhang, Henk Ellermann, Gosse Bouma, "System Description: A Case Study in Information Retrieval, Inference, and Visualization in the Semantic Web," LNCS 4519, pp.

769-778, Jun. 2007.

- [4] H. Stuckenschmidt, F. van Harmelen, A. deWaard, T. Scerri, R. Bhogal, J. van Buel, I. Crowlesmith, C. luit, A. Kampman, J. Broekstra, and E. van Mulligen, "Exploring large document repositories with RDF technology: the DOPE project," *Intelligent Systems, IEEE*, Vol.19, No.3, pp. 34-40, 2004.
- [5] Resource Description Framework(RDF), <http://www.w3.org/RDF/>.
- [6] Dave Buckett, "RDF/XML Syntax Specification," W3C Recommendation, Feb. 2004.
- [7] Ioanna Koffina, Giorgos Serfiotis, Vassilis Christophides, Val Tannen, Alin Deutsch, "Integrating XML Data Sources using RDF/S Schemas: The ICS-FORTH Semantic Web Integration Middleware (SWIM)," *Semantic Interoperability and Integration 2005*.
- [8] Zhuan Li, Yuanzhen Wang, "An Approach for XML Inference Control Based on RDF," LNCS 4080, pp. 338-347, Sep. 2006.
- [9] Jeremy J. Carroll, Patrick Stickler, "RDF Triples in XML," *Extreme Markup Languages Conference 2004*, Aug. 2004.
- [10] Michael K. Smith, Chris Welty, and Deborah L. McGuinness, "OWL Web Ontology Language Guide," W3C Recommendation, Feb. 2004. <http://www.w3c.org/Tr/2004/REC-owl-guide-20040210/>.
- [11] Jian Zhou, Li Ma, Qiaoling Liu, Lei Zhang, Yong Yu, Yue Pan, "Minerva: A Scalable OWL Ontology Storage and Inference System," *ASWC 2006*, pp. 429-443, Set. 2006.
- [12] Jeen Broekstra, and Arjohn Kampman, "Sesame: An Architecture for Storing and Querying RDF Data and Schema Information," LNCS 2342, pp. 54-68, Jun. 2002.
- [13] 우은미, 박명제, 정진완, "OWL 데이터 검색을 위한 효율적인 저장 스키마 구축 및 질의 처리 기법", *정보과학회논문지 : 데이터베이스 제34권, 제3호*, pp. 206-216, 2007. 06.
- [14] Hyunchul Jang, Youngil Kim, Dongwook Shin, "An Effective Mechanism for Index Update in Structured Documents," *ACM CIKM 1999*, pp. 383-390, Nov. 1999.
- [15] Kazuhito Kobayashi, Wenxin Liang, Dai Kobayashi, Akitsugu Watanabe, Haruo Yokota, "VLEI code: An Efficient Labeling Method for Handling XML Documents in an RDB," *ICDE 2005*, pp. 386-387, Apr. 2005.
- [16] Jingtao Zhou, Mingwei Wang, Shusheng Zhang, Hongwei Sun, "Semi-Structure Data Management by Bi-Directional Integration Between XML and RDB," *CSCWD 2006*, pp. 1077-1081, May. 2006.
- [17] Yuanbo Guo, "Univ-Bench Artificial data generator (UBA)," <http://swat.cse.lehigh.edu/projects/lubm/>.
- [18] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin, "LUBM: A Benchmark for OWL Knowledge Base

Systems," *Journal of Web Semantics*, Vol.3, No.2, pp. 158-182, 2005.



김진형

2004년 홍익대학교 컴퓨터공학과(공학사). 2006년 고려대학교 컴퓨터학과(이학석사). 2006년~현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 데이터베이스, XML, 유비쿼터스 컴퓨팅, 시맨틱 웹, 온톨로지

정동원

정보과학회논문지 : 데이터베이스
제 35 권 제 2 호 참조

백두권

정보과학회논문지 : 데이터베이스
제 35 권 제 2 호 참조