http://dx.doi.org/10.5392/JKCA.2016.16.02.374

RDF 그래프 패턴을 고려한 프로버넌스 압축 기법

Provenance Compression Scheme Considering RDF Graph Patterns

복경수*, 한지은*, 노연우*, 육미선*, 임종태*, 이석희**, 유재수*
충북대학교 정보통신공학과*, 동아방송예술대학교 뉴미디어콘텐츠과**

kyoungsoo Bok(ksbok@chungbuk.ac.kr)*, Jieun Han(jieun24@chungbuk.ac.kr)*,
Yeonwoo Noh(ywnoh@chungbuk.ac.kr)*, Misun Yook(misun@chungbuk.ac.kr)*,
Jongtae Lim(jtlim@chungbuk.ac.kr)*, Seok-Hee Lee(seoklee@dima.ac.kr)**,
Jaesoo Yoo(yjs@chungbuk.ac.kr)*

요약

프로버넌스 데이터는 데이터의 근원 정보나 변경 이력을 표현하는 메타데이터이다. 프로버넌스 정보는 변경 이력 정보가 쌓이면서 원본데이터와 비교하여 수십 배에 달하는 양을 차지한다. 따라서 대용량의 프로버넌스 데이터를 효율적으로 압축하기 위한 기법이 요구된다. 본 논문에서는 RDF 그래프 패턴을 고려한 프로버넌스 압축 기법을 제안한다. 제안하는 기법은 표준 PROV 모델을 기반으로 프로버넌스를 표현하고 텍스트 인코딩을 통해 프로버넌스 데이터를 숫자로 인코딩한다. 그래프 패턴을 이용하여 RDF 데이터와 프로버넌스 데이터를 압축한다. 제안하는 기법은 기존 프로버넌스 압축 기법과는 달리 시맨틱 웹상의 RDF 문서를 고려하여 프로버넌스 데이터를 압축한다. 압축률, 처리시간에 대한 성능 평가를 통해 제안하는 기법의 우수성을 증명한다.

■ 중심어 : | RDF | 프로버넌스 데이터 | 압축 | PROV 모델 |

Abstract

Provenance means the meta data that represents the history or lineage of a data in collaboration storage environments. Therefore, as provenance has been accruing over time, it takes several ten times as large as the original data. The schemes for effciently compressing huge amounts of provenance are required. In this paper, we propose a provenance compression scheme considering the RDF graph patterns. The proposed scheme represents provenance based on a standard PROV model and encodes provenance in numeric data through the text encoding. We compress provenance and RDF data using the graph patterns. Unlike conventional provenance compression techniques, we compress provenance by considering RDF documents on the semantic web. In order to show the superiority of the proposed scheme, we compare it with the existing scheme in terms of compression ratio and the processing time.

■ keyword : | RDF | Provenance | Compression | PROV Model |

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과(IITP-2015-H8501-15-1013), 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(No.2013R1A2A2A01015710), 2015년도 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 에너지인력양성사업으로 지원받아 수행한 인력양성 성과임(No. 20144030200450)

접수일자 : 2015년 11월 13일
수정일자 : 2015년 12월 07일
심사완료일 : 2015년 12월 07일
교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 컴퓨팅 기술 및 네트워크의 발달로 수많은 사용자들이 웹을 통해 대용량의 데이터를 급속도로 생산하고 공유하게 되었다. 웹 정보량이 폭발적으로 증가됨에 따라 웹 문서를 자동적으로 인식하고 검색하기 위한 필요성이 대두되었다. 컴퓨터가 문서의 의미를 이해하고 조작할 수 있는 차세대 웹 기술로 시맨틱 웹(semantic web)이 등장하였다. 시맨틱 웹은 분산 환경에서 리소스에 대한 정보와 자원 사이의 관계-의미 정보를 기계가 처리 할 수 있는 온톨로지 형태로 표현하고 이를 자동화된 기계가 처리하도록 하는 프레임워크이다[1][2]. 현재 시맨틱 웹 기반의 연구가 활발히 연구되고 있으며 이를 지원하기 위해 W3C에서 RDF(Resource Description Framework) 데이터 구조가 연구되었다. RDF는 웹상의 자원의 정보를 표현하기 위한 규격으로 이종의 데이터 간의 어의, 구문 및 구조에 대한 공통적인 규칙을 지원한다. RDF는 그래프로 표현되고 주어(subject), 술어(predicate) 및 목적어(object)인 트리플 구조로 구성되어 있다[3].

RDF 데이터는 대용량의 그래프로 구성되어 있으며 가 데이터가 증가함에 따라 데이터를 효율적으로 저장하는 것은 중요해졌다[14][15]. 데이터가 계속해서 생성되고 변경됨에 따라 어디서 왔는지, 누가 생성했는지, 어떻게 변화되었는지 등에 관한 정보를 관리하는 것이 필요하게 되었다. 사용 이력 데이터를 관리함으로써 어떤 사용자가 하는 행위를 파악할 수 있고 데이터가 어떻게 변하였는지도 알 수 있다. 이러한 데이터의 이력 정보를 관리하기 위한 메타데이터로 프로버넌스 데이터가 등장하였다. 프로버넌스 데이터는 데이터의 근원 정보나 사용 이력을 나타내는 메타데이터이다. 이러한 프로버넌스 데이터를 활용하여 사용자가 변경한 데이터 및 사용 이력을 파악할 수 있다[5].

프로버넌스 데이터를 관리하기 위한 표준 모델로 W3C에서는 PROV 모델을 제안하였다. PROV 모델은 객체(entity), 활동(activity), 에이전트(agent), 속성(property)으로 이루어져 있다[6]. 객체는 시맨틱 웹에 표현되는 RDF 문서를 나타낸다. 활동은 시맨틱 웹상의 문서를 변경하고 삭제하는 등 다양한 활동들을 나타낸다. 마지막으로 에이전트는 활동을 행하는 개인이나 조직을 나타내고 있다. 또한 표준 규격인 PROV 모델을 이용하므로 써 프로버넌스 데이터를 관리할 때 시맨틱 웹 데이터와의 호환성을 향상시킬 수 있으며 표준 질의 언어를 통해 검색이 가능하다. 표준 질의 언어로 SPARQL은 SQL과 유사한 질의 언어이며 RDF 데이터에 대한 질의를 위해 사용한다[8]. 그렇기 때문에 PROV 모델을 사용함으로써 SPARQL을 이용할 수 있다.

다수 사용자에 의해 RDF 데이터를 계속적으로 사용하고 변경함에 따라 프로버넌스 데이터가 지속적으로 생성됨에 따라 원본 데이터에 비해 수십 배의 대용량 데이터가 될 수 있다. 또한, 프로버넌스 내에는 많은 중복 정보들이 저장될 수 있다. 예를 들어, e-science는 국내외에 있는 연구원 정보와 연구 정보 등 다양한 연구 활동 정보를 인터넷 같은 사이버 공간에서 공동 활동으로 연구하는 차세대 연구 활동이다[4]. e-science는 다양한 분야에서 활용되고 있으며 점차 적용범위가 확대되고 있다. e-science에서는 혼자서 실험할 수 없는 대규모의 모의시험을 분산하여 시행하거나 상호 협력을 통해 하나의 주제를 분석하는 등 다양한 활동을 하게 된다. 이때 대규모의 모의시험을 공유하기 때문에 누가 어느 부분을 실험 하였는지 분석했는지 어떻게 수행했는지에 관한 정보들이 저장된다. 이러한 데이터가 프로버넌스 데이터에 해당한다. 프로버넌스 데이터는 지속적으로 쌓이게 되고 결국 원본 데이터의 관리보다 프로버넌스 데이터를 관리하는 것이 더 어려워진다. 또한 위키피디아는 국내외 사용자들이 자유롭게 정보를 올릴 수 있는 협업 저장소이다. 위키피디아에서는 하나의 문서를 여러 명의 사용자가 변경할 수 있다. 또한 한명의 사용자가 여러 문서를 생성, 변경, 삭제 등 다양한 활동을 할 수 있다. 이 경우 반복된 작업을 통해 프로버넌스 데이터가 수십 배에 달하는 데이터가 되며 이러한 이력정보를 나타내기 위해 그래프로 구성된다. 그렇기 때문에 그래프 압축이 필요할 뿐만 아니라 시맨틱 웹상에서는 데이터를 RDF데이터로 표현하기 때문에 RDF 데이터 압축 기법이 필요하다. 또한 프로버넌스는 사용자의 이력정보를 고려해야하기 때문에 프로버넌스의

흐름을 기반으로 한 프로버넌스 압축 기법이 필요하다.

최근 프로버넌스 데이터를 압축하거나 RDF 그래프를 압축 저장하기 위한 연구들이 진행되고 있다. 기존의 프로버넌스 관리 기법에서는 프로버넌스 데이터를 관리하기 위해 3개의 분해 기법과 2개의 상속기반의 함수를 제안했다[11]. 중복되는 부분은 분해하여 동일한 부분을 상속시켜서 효율적으로 저장한다. 프로버넌스 데이터를 압축하기 위해 웹 그래프 기반의 압축방법과 사전기반의 인코딩 기법을 결합시킨 프로버넌스 압축 기법을 제안하였다[13]. 일반적인 프로버넌스 압축 기법에서는 중복되는 부분을 압축하여 기존의 프로버넌스 데이터를 관리한다. 하지만 표준 프로버넌스 모델을 이용한 압축 기법은 없으며 일반적인 처리 데이터를 이용하여 압축하였기 때문에 RDF 프로버넌스 데이터에 적용하여 관리하기 힘들다. 또한, 기존의 프로버넌스 압축기법으로 압축하였을 때는 술어의 부분이 손실될 가능성이 있다.

본 논문은 프로버넌스 데이터를 표현하기 위해 확장된 PROV 모델을 이용하여 대용량의 RDF 프로버넌스 데이터를 관리하기 위한 압축 방법을 제안한다. 기존의 PROV 모델은 변경한 시간과 변경된 RDF 문서를 표현하지 못하기 때문에 기존의 PROV 모델을 확장한다. 또한 프로버넌스 데이터가 문자열로 표현되기 때문에 사전 인코딩을 통해 PROV 모델의 모든 데이터를 숫자 데이터로 저장한다. 사전 인코딩을 통해 문자열을 숫자 데이터로 저장함으로써 저장량을 감소시킨다. 또한 기존의 PROV 모델과는 달리 확장된 PROV 모델에서는 변경되거나 추가될 RDF 문서를 다루고 있다. 그래서 원본 RDF 압축을 통해 변경하거나 추가된 RDF 문서를 압축한다. 마지막으로 PROV 모델에서 활동 노드의 중복되는 부분은 서브그래프로 만들어 압축 저장함으로써 데이터의 사용이력을 고려하여 프로버넌스 데이터를 압축할 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 데이터 프로버넌스 연구 방향에 대해 기술하고 III장에서는 제안하는 프로버넌스 데이터 압축 기법을 기술한다. IV장에서는 다양한 환경에서 진행한 실험 환경을 기술한다. 마지막으로 V장에서 본 논문의 결론을 기술한다.

II. 관련연구

프로버넌스 데이터와 RDF 데이터를 효율적 저장하기 위한 기법들이 활발하게 연구되고 있다. 데이터 프로버넌스를 관리하는 대표적인 모델로 W3C에서 PRVO Model이 제안되었다[6]. PRVO 모델은 시스템 사이에 프로버넌스 정보를 교환할 수 있도록 지원하는 모델로 객체, 활동 및 에이전트의 관계로 구성되어 있다.

그림 1은 기본적인 PROV 모델을 나타낸다. 원은 객체(emtity)를 나타내며 데이터 사물이나 문서 같은 다양한 종류의 사물이다. 네모는 활동(activity)으로 실제 수행되는 부분이다. 오각형은 에이전트(agent)로 해당 활동을 처리하는 역할로 사람이나 소프트웨어 등을 나타낸다. 예를 들어 위키피디아의 변경이력을 PROV 모델로 만들 경우 객체는 위키피디아의 페이지에 해당하고 활동은 페이지의 추가, 삭제, 수정 등 활동을 뜻한다. 에이전트는 위키피디아의 페이지를 수정할 경우 수정하는 사람을 뜻한다. 'Used' 속성은 그래프에서 활동에서 객체로 연결하는 것으로 객체의 실행에 필요한 객체를 가리킨다. 'wasGeneratedBy' 속성은 객체에서 활동을 연결한 것으로 활동으로 인해 나온 결과물인 객체가 해당 활동을 가리킨다. 'wasDerivedFrom' 속성은 객체에서 객체를 연결하는 속성이다. 'wasInformedBy'는 하나의 활동에 의해 생성된 객체가 다른 객체와의 교환을 나타내는 속성이다. 'wasAttributedTo' 속성은 에이전트가 객체의 영향을 주는 것을 말한다.

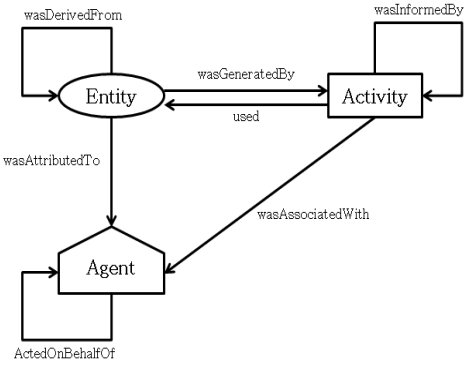


그림 1. PROV 기본 모델

'ActedOneBehalfOf' 속성은 에이전트가 특정 에이전트를 대신하는 것을 의미한다. 그 다음으로 에이전트와 활동을 연결하는 속성으로 'wasAssociatedWith'가 있다.

RDF 데이터 압축 기법으로 k^2-tree를 이용한 RDF 압축 기법을 제안하였다[8]. k^2-tree를 이용한 RDF 압축 기법은 RDF 데이터에서 술어 부분이 가장 적은 것을 이용하여 술어부분만 따로 인코딩하고 주어와 목적어를 함께 인코딩하는 방식으로 만약 주어와 목적어에 모두 포함되었을 경우 동일된 주어와 목적어는 함께 인코딩을 하고 동일한 부분을 제외한 나머지만 각 각 인코딩을 한다. 그 후 k^2-tree를 이용하여 술어마다 주어와 목적어를 트리로 구성한다.

RDF 압축을 위해 HDT를 이용한 압축 기법이 제안되었다[9]. [8]의 인코딩 방법과 유사하지만 인코딩 후 주어에 대해 그래프를 정렬한다. 술어와 목적어는 비트로 변환 되는데 이 때 술어의 경우에는 각 각의 부모 노드 즉, 주어에 대해 자식 노드가 여러 개인 경우 마지막 한 개만 1비트를 할당하고 마지막을 제외한 모든 노드는 모두 0비트를 할당한다. 목적어도 동일한 방식으로 술어가 부모 노드라고 생각하고 비트를 할당한다.

[10]에서는 패턴을 이용한 RDF 압축 방법을 제안하였다. RDF 데이터가 입력될 때 그래프들 중 술어는 동일하지만 주어와 목적어가 다른 경우가 있다. 이를 하나의 패턴으로 만들어 저장한다. 하나의 동일한 주어에 대해 여러 개의 술어가 동일하게 묶여있고 목적어만 다것이라는 가정 하에 하나의 패턴을 생성하게 된다. 이를 메모리에 저장하여 LRU(Least Recently Used) 전략을 이용하여 최근 사용이 가장 적은 것부터 교체한다.

프로버넌스 데이터의 저장 공간을 감소시키기 위해 프로버넌스 데이터 관리 기법을 제안되었다[11]. 중복되는 프로버넌스 데이터는 상속과 분해를 통해 관리한다. 이 기법에서는 3개의 분해 기법과 2개의 상속기반의 함수를 제안한다. 기본 분해에서는 동일한 부분을 하나의 복사 본만 Prov Store에 저장을 한다. 노드 분해에서는 동일하나 유사하고 동일한 부모와 자식을 가지고 있는 부분을 결합한다. 예를 들어 a, b, c 순서로 이루어진 그래프와 a, x, c 순서로 이루어진 그래프는 a, c가 동일하고 b와 x가 다르다면 a, b or x, c로 결합하게 된다. 인수 분해는 기본분해를 통해서만 분해할 경우 자식 노드가 동일하지 않을 경우 분해되지 않는 특성 때문에 프로버넌스 데이터에서 동일한 부분을 조각으로 나누어서 저장한다. 그리고 구조적 상속에서는 두 데이터가 부모 자식 관계를 가지고 있고 자식 데이터가 같은 프로버넌스 데이터를 가지고 있다면 자식 노드의 프로버넌스 데이터 정보는 기록되지 않는다. 마지막 술어 상속에서는 기존에 분해되었던 Prov Store에서 공통의 부분을 제외한 부분을 각 각 저장한다.

SPARQL 질의를 통해 동적으로 프로버넌스 데이터를 업데이트하는 모델을 제안하였다[12]. 이 모델에서는 named graph로 표현하는 동적은 프로버넌스 데이터를 이용하며 RDF의 변화를 삽입, 삭제, 로드등 기존의 SPARQL에 있는 질의 언어를 이용하므로 써 RDF 그래프의 프로버넌스를 저장하게 된다. 또한 이전 프로버넌스 데이터는 로그 기록으로 관리한다.

[13]에서는 웹 그래프 기반의 압축방법과 사전기반의 압축기법을 결합한 프로버넌스 압축 기법을 제안하였다. 데이터 프로버넌스가 웹 그래프와 유사하다는 점을 이용하여 중복된 데이터 프로버넌스 정보를 웹 그래프 기반의 압축 기법을 이용하여 압축하였다. 이 기법에서는 프로버넌스 데이터가 웹 그래프와 유사하게 지역성, 유사성, 연속성을 갖는다. 이를 이용하여 노드사이의 동일한 프로버넌스 데이터를 가질 경우 비트 리스트를 이용하여 압축하고 연속된 숫자는 RLE 인코딩을 통해 압축한다. RLE 인코딩이란 연속된 숫자가 발생할 경우 연속된 수만큼의 길이와 첫 번째 숫자만을 적용시킨다. 마지막으로 델타 인코딩을 통해 수의 차이만큼만 적용한다. 델타 인코딩을 적용하면 높은 자리수를 낮은 자리수로 변환 할 수 있다.

기존 기법에서는 효율적으로 RDF 데이터를 압축하거나 프로버넌스 데이터를 압축하는 기법들이 제안되었다. 프로버넌스 데이터를 압축하는 기법들은 동일한 부분을 압축 저장함으로써 데이터 저장량의 크기를 줄일 뿐만 아니라 사전 인코딩을 통해 문자열 데이터를 인코딩함으로써 저장 공간을 감소시킬 수 있고 RDF 압축에서는 술어의 개수가 적은 것을 이용하여 술어를 통한 압축기법을 제안함으로써 보다 중복을 줄일 수 있

다. 하지만 이를 결합한 압축 기법은 제안되지 않았다. 또한 시맨틱 웹상의 데이터는 대용량이기 때문에 표준 관리 모델을 사용함으로서 사용자들에게 높은 호환성을 가질 수 있다. 제안하는 기법에서는 PRVO 모델을 이용하여 데이터 프로버넌스의 변화를 관리할 뿐만 아니라 중복되는 프로세스를 이용하여 데이터 프로버넌스를 압축하여 저장한다.

III. 제안하는 프로버넌스 압축 기법

1. 제안하는 기법의 구조

프로버넌스 데이터는 원본데이터에 비해 수십 배에 달할 수 있다. 프로버넌스 데이터를 확인하기 위해서는 시간의 흐름이나 정보의 변경이력을 고려하여 압축하여야한다. 하지만 기존의 RDF 압축 기법은 변경이력을 고려하지 않았다. 또한, 기존의 프로버넌스 압축 기법들은 RDF 데이터에 대해서는 고려하지 않는다. 제안하는 기법은 확장된 PROV 모델을 사용함으로서 정보의 이력 정보를 고려하여 압축할 수 있다. 확장된 PROV 모델에는 시간이 표기되므로 시간의 흐름에 따라 변경된 것을 확인할 수 있기 때문이다.

[그림 2]는 제안하는 압축 처리 과정을 나타낸다. 데이터가 처음 삽입되면 사전 인코딩을 통해 모든 데이터를 문자열 데이터에서 숫자 데이터로 인코딩한다. 이 과정을 통해 숫자 데이터를 변경됨으로서 실제 데이터의 저장 공간을 감소시킨다. 이후 원본 RDF 압축을 통해 시맨틱 웹상의 RDF 문서를 압축을 한다. 원본 RDF 압축에서는 사전 인코딩 모듈과 동일하게 인코딩을 진행하지만 인코딩 방식이 다르다. 원본 RDF 압축에서는 주어와 목적어는 같이 인코딩하지만 술어는 따로 인코딩한다. 이때, 동일한 패턴의 RDF 문서가 나올 경우 이를 패턴으로 만들어 RDF 문서를 압축한다. 여기서 동일한 패턴이란 술어의 사용이 동일한 경우를 뜻한다. 마지막으로 프로버넌스 패턴 압축에서는 프로버넌스 그래프에서 PROV 모델의 활동 노드를 기준으로 서브그래프를 추출한다. 추출 후에는 추출된 서브그래프의 빈도수에 따라 일정 수치 값 이상이 나오면 패턴화를 한다. 패턴 화된 정보를 통해 최종 그래프를 변경한다.

그림 2. 제안하는 압축 처리 과정

2. PROV 모델의 확장

PROV은 프로버넌스 데이터를 관리하기 위해 W3C에서 제안된 표준 모델이다. PROV 모델은 시맨틱 웹에서 프로버넌스 데이터를 관리 하는 방법이 상이할 경우 호환성이 결여될 뿐만 아니라 대부분의 시맨틱 웹 데이터는 표준 규격인 PROV 모델로 표현이 가능하다. 제안하는 기법은 프로버넌스의 흐름을 나타내기 위해서 PROV 모델을 이용한 프로버넌스 압축 기법을 제안한다.

PROV 모델은 프로버넌스 데이터를 관리하기 위한 모델로 데이터의 흐름을 나타낸다. 기존의 PROV 모델은 기존의 프로버넌스 데이터를 표현하는데 용이하지만 웹상의 RDF 문서를 표현하는 노드기 없기 때문에 RDF 문서를 표현하기에는 부족하다. 또한, 시간에 흐름에 따라 작성되지만 언제 변경하였는지에 관한 정확한 정보가 표시하지 않는다. 따라서 기존의 PROV 모델을 확장하여 메타데이터를 표현하는 부분을 추가한다. 메타데이터를 통해 기존 모델과는 달리 시맨틱 웹상의 RDF 문서의 변경 부분 및 변경된 시간을 알 수 있다.

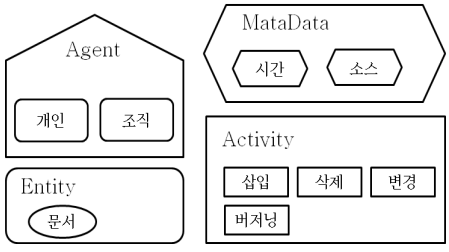


그림 3. PROV 모델 확장

[그림 3]은 확장된 PROV 모델을 나타낸 것이다. 기존의 PRVO 모델에서 메타데이터를 추가하여 언제 데

이터가 변형되었는지 무슨 RDF 문서가 변형하였는지에 관한 정보를 표현한다.

표 1. 제안하는 모델의 요소 정의

클래스	서브클래스	설명
객체	문서	RDF으로 구성된 문서
에이전트	개인	활동 행하는 개인
	조직	활동 행하는 조직
활동	삽입	기존의 문서에서 RDF 데이터가 삽입될 때
	삭제	기존의 문서에서 RDF 데이터가 삭제될 때
	변경	기존의 RDF 문서가 새로운 RDF 문서로 변형될 때
	버저닝	기존의 RDF 문서가 새롭게 버저닝 될 때
메타데이터	시간	활동이 행해지는 시간
	소스	활동에 의해 추가, 삭제, 변경될 RDF 데이터

[표 1]은 제안하는 모델의 요소들을 정의한 것이다. 에이전트는 개인과 조직으로 나누어지며 실제 활동을 동작하는 주체에 해당한다. 객체는 문서로 RDF로 구성된 문서를 의미한다. 활동은 삽입, 삭제, 변경 및 버저닝으로 총 4가지로 구성된다. 삽입은 메타데이터는 실제 활동이 동작할 때 생성되고 시간이나 변경할 문서를 나타낸다. 예를 들어, 위키피디아의 변경 이력들을 PRVO 모델로 만든다면 객체들은 위키피디아의 페이지를 뜻하고 에이전트는 페이지를 변경하는 개인을 뜻한다. 활동은 그 페이지에 내용을 추가하거나 새로운 페이지를 생성하는 활동들을 가리킨다. 메타데이터에서 시간이란 그 페이지를 수정하거나 추가한 시간을 뜻하고 소스란 변경하였을 경우에는 변경한 내용, 새로 페이지를 추가하였을 때는 새로 추가된 내용을 뜻한다.

[그림 4]는 기존의 PROV 모델을 이용한 예제이다. 기존의 문서 C와 문서 D를 삽입하여 새로운 문서 F를 생성하였고 그것은 지은이라는 개인에 의해 만들어진다. 문서 F는 선화에 의해 삽입되어 새로운 문서 X를 생성하였다. 이 경우 어떤 부분을 변경하였고 언제 변경하였는지 알 수 없다.

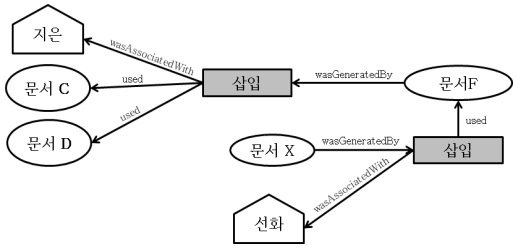


그림 4. 기존 PROV 모델 예제

확장된 PROV 모델은 언제, 어떤 부분을 변경하였는지 알 수 있다. [그림 5]는 확장된 PROV 모델을 이용한 예제이다. 문서 C와 문서 D를 삽입하여 새로운 문서 F를 생성하였고 '지은'이라는 개인에 의해 만들어진다. 2015년 9월 2일에 만들어졌으며 다음과 같은 RDF데이터를 추가하였다. 또한, 에이전트 선화는 새로 삽입된 문서 F를 이용하여 추가적인 RDF 데이터를 추가하여 새로운 문서 x를 생성하였다.

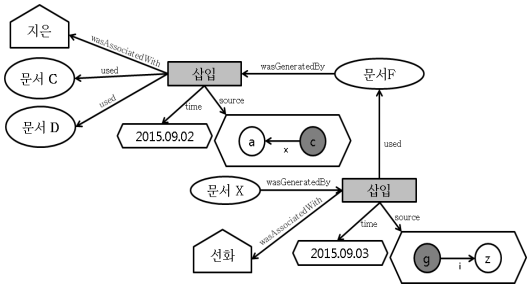


그림 5. 변형된 PROV 모델 예제

3. 사전 인코딩

데이터 프로버넌스는 원본 데이터에 비해 수십 배에 달하는 대용량 데이터로 구성되며 이 또한 문자열 데이터로 이루어져 있다. 예를 들어, 위키피디아는 하나의 페이지를 여러 명의 사용자가 변경하여 관리한다. 따라서 실제 데이터를 문자열 데이터로 저장할 경우 많은 공간을 차지한다. 대부분의 인코딩은 문자열 데이터를 숫자 데이터로 변경한다. 사전 인코딩 모듈에서는 입력된 데이터를 분석하여 각 각의 노드들과 가지들을 인코딩한다. 이때 에이전트, 메타데이터, 객체 노드를 인코딩한 데이터 테이블과 활동노드를 인코딩한 활동 테이

블 속성을 인코딩한 술어 테이블로 총 3개의 테이블에 인코딩한다.

사전 인코딩에서는 입력된 프로버넌스 데이터를 분석하여 텍스트 인코딩을 통해 데이터를 인코딩한다. 텍스트 인코딩은 기존의 텍스트 인코딩과는 달리 3개의 테이블에 각 각의 데이터를 관리한다. 텍스트 인코딩은 입력된 순서로 인코딩되며 처음 데이터가 입력되면 데이터를 분석하여 기존의 인코딩 테이블에 인코딩된 데이터가 있는지 확인한다. 확인 후 데이터가 없으면 각각의 테이블에 구별되어 술어와 활동, 데이터를 인코딩한다.

[그림 6]은 인코딩 전의 PROV 모델이다. 기존 문서B에 RDF 문서를 삽입하여 새로운 문서 A를 생성하였다. 다음과 같은 PROV 모델이 입력되면 인코딩 모듈에서는 인코딩 테이블을 검색한다. [표 2]는 인코딩 테이블이다. 만약 문서 A를 인코딩을 하면 우선적으로 인코딩 테이블에서 데이터를 확인한 후 해당 데이터가 없을 경우 새로 ID를 부여한다. 새로운 ID로 인코딩 할 때는 |ID+1|을 한다.

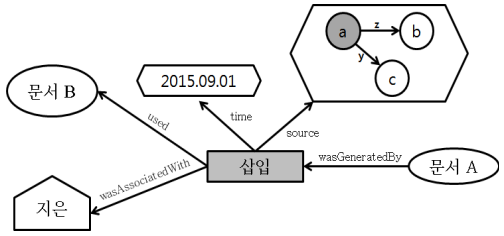


그림 6. 인코딩 전 PROV 모델

표 2. 인코딩 테이블

ID	String	
1	문서 B	data
2	2015.09.01	
1	변경	Activity
2		
1	used	Predicate
2	wasAssociatedWith	

인코딩된 데이터는 그래프와 인코딩에 반영된다. [그림 7]은 텍스트 인코딩을 통해 변경된 PROV 모델이다. 사전 인코딩 모듈에서는 텍스트 인코딩을 통해 문자열을 숫자로 인코딩하므로써 프로버넌스 데이터의 저장량을 감소시킨다.

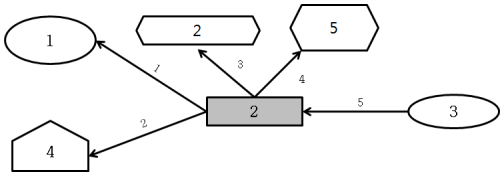


그림 7. 인코딩 후 PROV 모델

4. 원본 RDF 압축

확장된 PROV 모델에서는 변경할 RDF 원본 데이터도 관리하고 있다. 이에 따라 RDF 원본 데이터가 대용량일 경우 많은 저장 공간을 차지하므로 압축을 수행한다. 또한, 기존의 RDF 데이터는 주어와 목적어에 비해 술어의 양이 적다. 이를 이용하여 RDF 원본 데이터에서 술어를 기준으로 동일한 패턴을 가진 RDF 그래프는 패턴으로 만들어서 해당 패턴에 포함된 변수는 변수 테이블을 만들어 관리한다.

그림 8. 원본 RDF 압축 과정

[그림 8]은 원본 RDF 압축의 전체적인 과정을 나타낸 것이다. 처음 메타데이터에서 소스가 가리키고 있는 원본 RDF문서가 있다. 이는 시맨틱 웹상의 문서를 뜻한다. 처음 메타데이터에 의해 원본 RDF 문서가 입력되면 RDF 인코딩을 통해 문자열 데이터가 숫자 데이터로 변경된다. 이는 사전인코딩 모듈과 동일하지만 인코딩 방식이 다르다. 사전 인코딩에서는 입력된 순으로 순차적으로 3개의 인코딩 테이블에 인코딩되지만 RDF 인코딩에서는 주어와 목적어는 동일한 인코딩 테이블에 인코딩하고 술어만 따로 인코딩을 하게 된다. 인코딩된 원본 RDF 문서는 RDF 패턴 압축을 통해 압축된다. RDF 패턴 압축에서는 주어와 목적어의 수보다 술

어의 수가 적다는 RDF 특성을 이용하여 동일한 술어를 사용한 경우 패턴으로 만들어 압축 저장하게 된다.

[그림 9]는 원본 RDF 데이터이다. RDF 데이터는 그래프로 구성되어 있으며, RDF 데이터의 대부분은 문자열로 구성된다. [그림 9]에서는 주어가 A, 술어가 D, 목적어가 C가 된다. 처음 RDF 데이터가 주어와 목적어는 같이 인코딩하고 술어만 구별하여 인코딩을 한다.

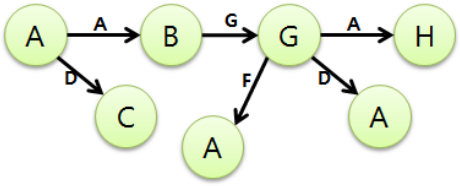


그림 9. 원본 RDF 데이터

처음 원본 데이터가 입력되면 RDF 원본 데이터 인코딩 테이블에서 해당하는 인코딩 ID가 있는지 검색한다. 만약 인코딩 테이블에 존재하지 않을 경우 마지막 ID에서 더하기 1을 하여 인코딩한다. [표 2]는 [그림 7]의 RDF 데이터를 인코딩한 예이다. 주어와 목적어에 해당하는 것과 술어에 해당하는 것을 각각 인코딩한다.

표 3. RDF 원본 데이터 인코딩 테이블

ID	String	
1	A	SO
2	B	
3	G	
4	C	
5		
1	A	P
2	D	
3	F	
4	G	

[그림 10]은 RDF 데이터 인코딩을 반영한 RDF 원본 데이터 그래프이다. 예를 들어, A가 기존의 테이블에 존재할 경우 기존의 테이블에 있는 것을 가져와 사용하지만 H와 같이 기존의 테이블에 없을 경우 마지막 ID 4에서 더하기 1을 하여 ID 5로 인코딩된다.

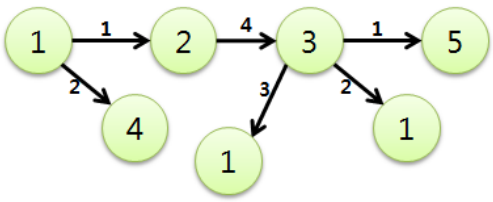


그림 10. 인코딩된 원본 RDF 데이터

RDF 데이터는 동일한 패턴의 술어를 가지는 경우가 있다. 동일한 패턴이 나온 경우 주어와 목적어를 변수로 두어 패턴을 추출한다. [그림 11]은 RDF 데이터에서 추출된 RDF 패턴이다. [그림 10]에서는 술어 1와 술어 2가 동일하게 사용된다. 그렇기에 변수의 경우 테이블로 저장한다. [표 4]는 RDF 패턴에서 나온 변수와 ID 값이다. 다음과 같이 테이블로 RDF 패턴 변수를 저장한다.

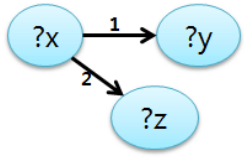


그림 11. RDF 패턴

표 4. RDF 패턴1의 변수 테이블

변수	순서	ID
?x	1	1
	2	3
?y	1	2
	2	5
?z	1	4
	2	1

[그림 12]는 [그림 10]의 RDF 패턴 기준으로 압축된 RDF 문서이다. 각 각의 패턴에 해당하는 데이터는 변수 테이블에 저장 되고 RDF 문서에는 압축된 그래프로 나타낸다. [그림 10]의 [1 1 2]와 [1 2 4]가 동일하게 사용되므로 압축하여 [표 4]와 같이 변수 테이블에 저장하고 [그림 12]과 같이 패턴 압축을 반영한다.

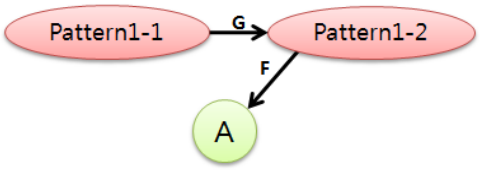


그림 12. RDF 원본 데이터 패턴 압축 결과

5. 프로버넌스 패턴 압축 기법

프로버넌스 데이터를 처리하는 패턴은 동일하게 반복되는 경우가 많다. 예를 들어, 문서 사용의 패턴을 보면 그 문서를 생성한 후 사용자들이 사용하다가 필요한 부분을 변경 하는 등 여러 가지의 문서에 대해 유사하거나 동일한 사용 패턴을 보인다. 이를 이용하여 프로버넌스 패턴 압축모듈에서는 반복되는 사용 패턴을 추출하여 압축 저장한다.

[그림 13]은 프로버넌스 패턴 압축 모듈의 전체적인 과정이다. 처음 사전 인코딩에서 숫자데이터로 변경된 PROV 모델이 입력되면 PROV 모델에서 활동 노드가 동일하게 반복하는 서브그래프를 추출한다. 예를 들어, 문서의 이력 중 삽입이라는 활동 뒤에 항상 변경이 자주 이러나면 삽입과 변경 순으로 표현된 PROV 모델을 서브그래프로 추출한다. 추출된 서브그래프가 일정 수치 값 이상 나올 경우 기준 패턴이라 명하고 이를 압축하여 저장한다.

그림 13. 프로버넌스 패턴 압축 과정

[그림 14]는 프로버넌스 데이터 그래프에서 서브그래프를 추출하는 과정을 나타낸다. [표 5]와 같이 서브그래프가 생성되는데 sub1 과 sub10이 동일한 그래프이기 때문에 sub10은 sub1로 변경된다. 계속해서 패턴을 추출하다가 최근에 사용되지 않는 패턴은 메모리에서 삭제한다.

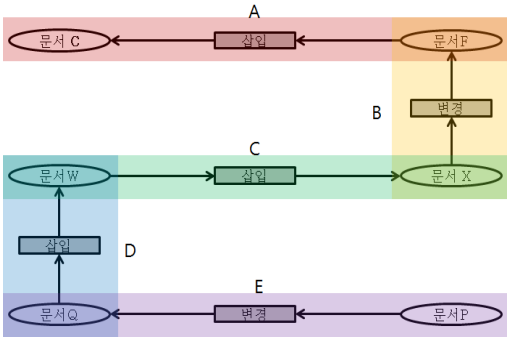


그림 14. 서브 그래프 생성

표 5. 서브그래프

Sub graph	
sub1	A+B
sub2	A+B+C
sub3	B+C
sub4	A+B+C+D
sub5	B+C+D
sub6	C+D
sub7	A+B+C+D+E
sub8	B+C+D+E
sub9	C+D+E
sub10	D+E

서브그래프가 생성된 뒤 각 각의 서브그래프의 횟수 정보를 통계 테이블로 관리한다. [표 6]는 서브 그래프의 통계 테이블이다. 통계 테이블에 따라 일정 수치가 초과 할 경우 기준 패턴으로 압축 저장한다.

표 6. 서브그래프 통계 테이블

sub	num
sub1	2
sub2	1
sub3	1
sub4	1

[그림 15]는 제안하는 기법에 따라 패턴 압축된 RROV 모델이다. 제안하는 기법에서는 추출된 서브그래프 중 동일한 서브그래프가 나오면 반복되는 서브그

래프를 기준 패턴으로 하여 저장한다. [그림 14]에서와 같이 sub1과 sub10이 동일하므로 기준 패턴으로 생성되며 [표 7]과 같이 스트링 데이터로 변환되어 저장된다. 최종 결과는 기준 패턴으로 변환된 노드로 저장하여 프로버넌스 데이터의 그래프를 압축 저장한다.

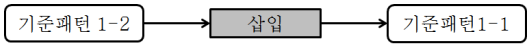


그림 15. 패턴 압축

표 7. 기준 패턴 테이블

기준패턴 1-1	기준 패턴 1-2
문서C/문서F/문서X	문서W/문서Q/문서P

IV. 성능평가

제안하는 문서 이력 정보 관리를 위한 RDF 프로버넌스 관리 기법과 기존 기법과의 성능 비교 평가를 통해 제안하는 기법의 우수성을 입증한다. 제안하는 기법에서 사용한 실험 환경은 [표 8]과 같다. 실험데이터는 문서의 이력 정보를 관리하기 위한 PROV 모델을 구현하였다. 데이터는 RDF 문서를 관리하기 위한 PROV 모델을 기반으로 소스에는 각각의 RDF 문서와 변경한 시간이 나타나고 각 각의 객체에 해당 문서의 이름이 표기된다. 활동 노드는 각각의 문서를 수정하는 행위를 표기하였으며 에이전트는 해당 활동의 행위자로 각 각 활동 20개와 객체 21개, 에이전트 20개, 소스 40개로 구성되어있다. JAVA 언어를 통해 압축 기법을 구현하고 성능을 평가하였다.

표 8. 성능평가 환경

항목	값
CPU	Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz
RAM	4.0 GB
사용언어	JAV(TM) SE Runtime Euntime Environment

제안하는 기법은 RDF 문서는 기존의 RDF 압축 기법을 변형하였으며 프로버넌스 압축 기법은 유사한 패턴이 생성될 경우 빈도수를 고려하여 압축하였다. 기존 기법은 웹 기반 압축 기법으로 표기하였으며 웹 데이터 그래프에 맞게 제안되었음으로 RDF 프로버넌스에 맞게 변형하여 실험하였다[13]. 성능 평가는 압축률, 처리 시간, 정확성 관점에서 수행된다.

원본 데이터의 크기를 기준으로 하여 제안하는 기법과 웹 기반 압축 기법과의 성능 비교를 하였다. [그림 16]은 제안하는 기법과 웹 기반 압축 기법의 압축률을 비교하였다. 압축률은 원본데이터의 크기와 비교하여 계산되며, 식 (1)과 같다. 그 결과 제안하는 기법은 원비 데이터에 비해 50%가 압축되었고 웹 기반 압축 기법에 비해 8% 더 압축되었다.

$$\text{압축률} = \frac{\text{압축된 크기}}{\text{원본 데이터의 크기}} \times 100 \quad (1)$$

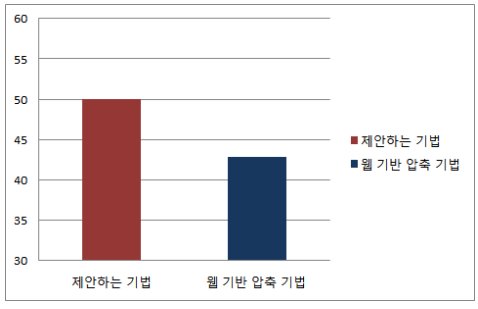


그림 16. 압축률

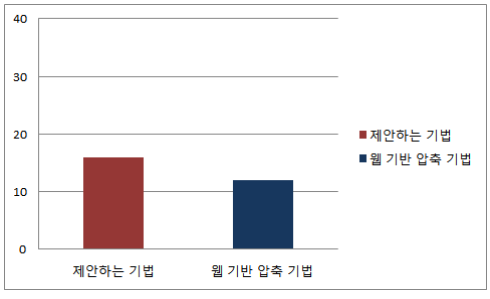


그림 17. 처리 시간

[그림 17]은 제안하는 기법과 웹 기반 압축 기법의 압축 처리 시간을 비교한 결과이다. 제안하는 기법과 웹 기반 압축 기법은 프로버넌스 데이터를 압축하기 이전에 동일하게 사전 인코딩한다. 하지만 프로버넌스 데이터만 압축하는 웹 기반 압축 기법과는 달리 제안하는

기법에서는 추가적으로 RDF 문서 압축하기 때문에 다소 시간의 차이를 나타낸다. 제안하는 기법은 웹 기반 압축 기법에 배해 4초 더 걸린 것을 확인할 수 있다.

압축 결과의 손실을 확인하기 위해 다양한 질의 유형으로 성능평가를 하였다. [표 9]은 정확성을 비교하기 위한 질의유형이다. 총 3가지로 구성되었으며 특정 에이전트와 활동에 대한 질의로 구성되어있다.

표 9. 질의 유형

Q	질의유형
Q1	특정 에이전트가 수정한 문서
Q2	특정 에이전트가 한 활동
Q3	특정 활동을 한 에이전트

[그림 18]은 [표 9]의 질의 유형을 기반으로 제안하는 기법과 웹 기반 압축 기법의 정확성을 비교하였다. 웹 기반 압축 기법은 RDF 기반의 프로버넌스 압축 기법이 아니기 때문에 압축 시 데이터가 손실되어 해당 질의 결과를 찾을 수가 없었다. 반면 제안하는 기법은 원본 데이터와 동일하게 모든 질의 유형을 찾을 수 있었다.

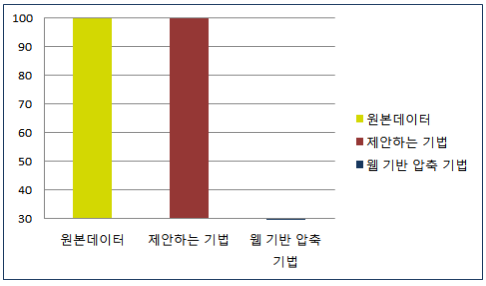


그림 18. 정확성

V. 결론

본 논문에서는 프로버넌스 데이터를 효율적으로 관리하기 위한 압축 기법을 제안했다. 제안하는 압축 기법은 PROV 모델을 이용하여 사전에 모든 데이터를 숫자로 인코딩하여 저장 공간을 감소시킨다. 또한, RDF 데이터에서 반복되는 부분은 RDF 패턴으로 만들어 저장하여 저장 공간을 감소시키고 기존에 반복되는 활동은 기준 패턴으로 만들어 중복되는 부분을 감소시켰다. 본 연구는 위키피디아 또는 e-science와 같은 협업 저장소에서 다수 사용자에 의한 문서 변경 내역을 관리하기 위해 사용될 수 있다. 성능평가 결과 제안하는 기법을 이용하여 원본데이터의 크기를 50% 감소시켰으며 정확성 또한 원본데이터와 동일하였다.

참 고 문 헌

[1] T. Berners Lee, J. Hendler, and O. Lassila, “The Sementic Web,” In Proceedings of the Scientific American, Vol.284, No.5, pp.34-43, 2001.

[2] Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, “The Semantic Web: The Roles of XML and RDF,” Journal of IEEE : Internet Computing, Vol.4, No.5, pp.63-73, 2000.

[3] http://www.w3.org/TR/rdf11-concepts/

[4] 안윤선, 김윤희, "과학 계산 실험을 위한 클라우드 자원을 활용한 실험 프로비넌스 모델 설계," 한국정보과학회 2014 한국컴퓨터종합학술대회 논문집, pp.1548-1550, 2014.

[5] 신은영, 이석훈, 백두권, “An Ontology Provenance Model for an Ontology Repository,” 정보과학회 논문지: 데이타베이스, 제41권, 제3호, pp.181-191, 2014.

[6] http://www.w3.org/TR/prov-overview/

[7] http://www.w3.org/TR/sparql11-query/

[8] S. Álvarez-García, N. R. Brisaboa, J. D. Fernández, and M. A. Martínez-Prieto, “Compressed k2-Triples for Full-In-Memory RDF Engines,” Association for Information Systems, 2011.

[9] J. D. Fernández, M. A. Martínez-Prieto, C. Gutiérrez, A. Polleres, and M. Arias, “Binary RDF representation for publication and exchange (HDT),” Journal of Web Semantics,

Vol.19, pp.22-41, 2013.

[10] N. F. García, J. Arias-Fisteus, L. Sánchez, D. Fuentes-Lorenzo, and Ó. Corcho, "RDSZ: An Approach for Lossless RDF Stream Compression," European Semantic Web Conference, pp.52-67, 2014.

[11] A. Chapman, H. V. Jagadish, and P. Ramanan, "Efficient provenance storage," Special Interest Group on Management of Data, pp.993-1006, 2008.

[12] H. Halpin and J. Cheney, "Dynamic provenance for SPARQL updates using named graphs," In Workshop on the Theory and Practice of Provenance TaPP-11, 2011.

[13] Y. Xie, K. Muniswamy-Reddy, D. Feng, Y. Li, and D. D. E. Long, "Evaluation of a hybrid approach for efficient provenance storage," TOS, Vol.9, No.4, p.14, 2013.

[14] 김기연, 윤종현, 김천중, 임종태, 복경수, 유재수, "대규모 RDF 데이터의 특성을 고려한 효율적인 색인 기법," 한국콘텐츠학회논문지, 제15권, 제1호, pp.9-23, 2015.

[15] 고훈준, 유원희, "응용프로그램의 검색을 위한 RDF 메타데이터 시스템의 설계," 한국콘텐츠학회논문지, 제5권, 제6호, pp.1-9, 2005.

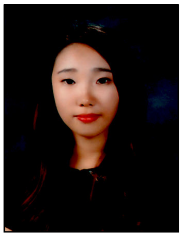
저 자 소 개

복 경 수(KyoungSoo Bok) 종신회원

- 1998년 2월 : 충북대학교 수학과(이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 전산학과 Postdoc
- 2008년 3월 ~ 2011년 2월 : ㈜가인정보기술 연구소
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 초빙교수

<관심분야> : 데이터베이스 시스템, 이동객체 데이터베이스, 소셜 네트워크, 빅데이터 등

한 지 은(Jieun Han) 준회원



- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 빅데이터, 프러버넌스 데이터, RDF 등

노 연 우(Yeonwoo Noh) 준회원

- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스 시스템, 빅데이터 등

육 미 선(Misun Yook) 준회원

- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 빅데이터, 분산처리시스템, 소셜 네트워크 서비스 등

임 종 태(Jongtae Lim) 정회원

- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2015년 8월 : 충북대학교 정보통신공학과(공학박사)

<관심분야> : 데이터베이스 시스템, 시공간 데이터베이스, 위치기반 서비스, 모바일 P2P 네트워크, 빅데이터 등

이 석 희(Seok-Hee Lee) 종신회원

- 1994년 2월 : 충북대학교 정보통신공학과(공학사)
- 1998년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2001년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2000년 3월 ~ 현재 : 동아방송예술대학교 뉴미디어콘텐츠과 부교수

<관심분야> : 정보검색, 데이터베이스, 이동객체, SNS, 센서네트워크 등

유 재 수(Jaesoo Yoo) 종신회원

- 1989년 2월 : 전북대학교 컴퓨터공학과(공학사)
- 1991년 2월 : KAIST 전산학과(공학석사)
- 1995년 2월 : KAIST 전산학과(공학박사)
- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과(전임강사)
- 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수
- 2009년 3월 ~ 2010년 2월 : 캘리포니아주립대학교 방문교수

<관심분야> : 데이터베이스시스템, 빅데이터, 센서 네트워크 및 RFID, 소셜 네트워크 서비스, 분산 객체컴퓨팅, 바이오인포매틱스 등