情報保護學會論文誌
第18卷 第2號, 2008. 4

RDF 웹 문서의 부분적인 정보 은닉과 관련한 접근 권한 충돌 문제의 분석*

김 재 훈†, 박 석
서강대학교 컴퓨터공학과

Analysis of Access Authorization Conflict for Partial Information Hiding of RDF Web Document*

Jaehoon Kim†, Seog Park
Sogang University, Dept. of Computer Science and Engineering

요 약

RDF는 W3C의 시맨틱 웹에서 사용하는 기본적인 온톨로지 모델이다. 그리고 더욱 다양한 온톨로지 관계를 정의하는 OWL은 이러한 RDF 기본 모델을 확장한 것이다. 최근 Jain과 Farkas는 RDF에 대한 RDF 트리플에 기반을 둔 접근 제어 모델을 제시하였다. 그들 연구의 초점은 RDF 온톨로지 데이터에서 고려해야 하는 추론에 의한 접근 권한 충돌 문제를 소개한 것이다. 비록 RDF 모델이 XML로 표현되지만, 기존의 XML 접근 제어 모델을 RDF에 적용하기 어려운 것이 바로 이러한 RDF 추론 때문이다. 하지만, Jain과 Farkas는 그들의 연구에서 먼저 RDF 접근 권한 명세시의 권한 전파가 RDF 상/하위 온톨로지 개념에 대하여 어떻게 이루어지는지를 정의하고 있지 않다. 이것이 중요한 이유는 추론에 의한 권한 충돌의 문제는 결국 권한 명세시의 권한 전파와 권한 추론시의 권한 전파 사이에서의 충돌 문제이기 때문이다. 본 논문에서는 먼저 RDF 트리플에 기반을 둔 RDF 접근 권한 명세 모델에 대하여 자세히 소개한다. 다음으로 이러한 모델을 바탕으로 RDF 추론 시의 권한 충돌 문제를 자세히 분석한다. 다음으로 권한 명세시의 권한 충돌 여부를 신속히 조사하기 위하여 포함 관계 추론과 관련한 그래프 레이블링 기법을 이용하는 방법을 간략히 소개한다. 마지막으로 Jain과 Farkas 연구와의 비교 및 제안된 충돌 발견 알고리즘의 효율성을 보이는 몇 가지 실험 결과를 제시한다.

ABSTRACT

RDF is the base ontology model which is used in Semantic Web defined by W3C. OWL expands the RDF base model by providing various vocabularies for defining much more ontology relationships. Recently Jain and Farkas have suggested an RDF access control model based on RDF triple. Their research point is to introduce an authorization conflict problem by RDF inference which must be considered in RDF ontology data. Due to the problem, we cannot adopt XML access control model for RDF, although RDF is represented by XML. However, Jain and Farkas did not define the authorization propagation over the RDF upper/lower ontology concepts when an RDF authorization is specified. The reason why the authorization specification should be defined clearly is that finally, the authorizatin conflict is the problem between the authorization propagation in specifying an authorization and the authorization propagation in inferencing authorizations. In this article, first we define an RDF access authorization specification based on RDF triple in detail. Next, based on the definition, we analyze the authoriztion conflict problem by RDF inference in detail. Next, we briefly introduce a method which can quickly find an authorization conflict by using graph labeling techniques. This method is especially related with the subsumption relationship based inference. Finally, we present a comparison analysis with Jain and Farkas' study, and some experimental results showing the efficiency of the suggested conflict detection method.

Keywords : RDF data, Ontology, Inference, Access Control, Authorization Conflict

I. 서 론

앞으로의 웹의 구조는 현재의 사용자들의 멀티미디어 기반의 정보공유에서 소프트웨어 에이전트들이 직접 웹 문서의 정보를 이해하고 활용하는 단계로 발전할 것이다. 현재 이러한 기대가 웹 2.0, 혹은 시맨틱 웹이라는 새로운 연구 영역을 통하여 반영되어 지고 있다.

RDF (Resource Description Framework) [1]와 OWL (Web Ontology Language) [2]은 현재 W3C에서 개발하고 있는 이러한 웹의 의미 표현을 위한 웹상의 표준 온톨로지 언어이다. RDF와 OWL의 관계는 OWL이 RDF 모델에 기반을 두고 있으며, RDF가 가지고 있는 단순한 온톨로지 표현 어휘에서 보다 풍부한 온톨로지의 관계를 정의할 수 있는 별도의 어휘를 제공한다. 이러한 RDF와 OWL 모델은 XML (eXtensible Markup Language) 언어로 기술된다.

웹 문서에 포함된 RDF 데이터에 대한 접근제어 (access control)를 고려할 때, RDF 데이터가 XML로 기술되기 때문에, 기존의 활발히 연구된 XML 접근 제어 모델 [3, 4, 5]의 적용을 기대할 수 있다. 하지만 이러한 단순한 접근은 최근 RDF 접근 제어 모델로 연구 되어진 몇몇 연구들 [6, 7, 8]에서 언급 되듯이, XML은 단지 RDF 모델을 표현하기 위한 기반 언어이기 때문에 바람직하지 않다. 그 가장 중요한 이유는 온톨로지와 관련한 추론 (inference) 때문이다. 즉, 웹 문서상의 RDF 데이터에 대하여 XML 접근 권한 (access authorization)을 명시적으로 기술한다 하더라도, 추론에 의하여 새롭게 생성되는 정보에 대해서는 어떤 접근 권한을 적용할지 XML 접근 제어 모델은 고려하고 있지 않기 때문이다.

1.1. 연구동기 및 의의

최근 Jain과 Farkas는 참고 문헌 [6]의 연구에서 RDF 데이터 추론에서의 RDF 트리플 (triple) 기반의 접근 제어 모델을 소개하였다. 이러한 연구는, 기존 RDF 접근 제어에 관한 몇몇 연구들 [7, 8, 9, 10]과 달리, RDF 모델의 기본적인 트리플 구조로 보안 객체 (security object)를 표현하고, 추론에 의한 접근 권한 충돌 (conflict)을 다루고 있기 때문에 큰 의미를 가진다고 생각한다.

하지만 Jain과 Farkas는 그들의 연구에서 RDF 트리플 기반의 접근 권한을 보안 관리자가 어떻게 명세하며, 명세된 접근 권한의 의미가 어떻게 해석될 수 있는 지를 설명하고 있지 않다. 단지 이미 할당된 접근권한에 대한 추론에 의한 충돌 해결 문제를 다룬다. 하지만, 결국 권한의 충돌은 권한 명세 (specifying authorization)와 권한 추론 (inferencing authorization) 사이의 충돌이기 때문에, 먼저 권한 명세의 방법과 의미를 정의하는 것은 매우 중요한 문제라고 생각한다. 본 논문에서는 XML의 트리 구조에 기반한 하위 노드로의 접근 권한 전파 명세 [3]와 같이 온톨로지 계층 구조 (ontology hierarchy)에 기반한 하위 개념 혹은 하위 속성으로의 권한 전파를 고려한 권한 명세 모델을 정의한다. 다음으로 RDF의 subClassOf, subPropertyOf, 도메인 추론 등의 세 가지 주요한 추론에 대하여 추론시의 권한 충돌 문제를 권한 명세와 연관 지어 자세히 분석한다. 다음으로 분석된 추론 시의 충돌 조건에 기반하여, 그래프 레이블링 기법 [11, 12]을 이용한 효율적 권한 충돌 발견 방법에 대하여 간단히 소개한다. Jain과 Farkas 또한 그들의 연구에서 권한 충돌 발견 방법을 소개하였지만, 그들이 제안한 방법은 매우 비효율적이다. 왜냐하면 제안된 알고리즘은 권한 충돌을 발견하기 위하여, 모든 RDF 인스턴스 데이터를 조사하기 때문이다. 즉, 새로운 접근 권한이 명세될 경우 먼저 해당 RDF 트리플들에 새로운 접근 권한의 보안 등급 (security label)을 할당한다. 다음으로 추론에 의해 모든 RDF 트리플에 대한 권한 전파 (propagating authorization)를 수행하며, 만약 어떤 추론 과정 중 권한 충돌이 발생했다면, 그 추론은 취소되고 접근 권한 또한 받아들여 지지 않는다. 이러한 방법은 RDF 데이터가 작은 규모가 아닐 때 비효율적일 것이다.

연구에서는 특별히 이러한 RDF 트리플 기반의 접근 권한 모델을 웹에서의 RDF 문서의 세밀한 (fine-grained) 정보 은닉 (information hiding)에 적용하는 것을 고려하여 소개한다. 즉, 어떤 사용자가 웹 브라우저를 통해 RDF 데이터를 브라우징할 경우, 자신에게 명시적으로 불허된 또한 불허된 추론을 일으킬 수 있는 데이터를 웹 브라우저에 보이지 않게 (invisible) 하는 것이다. 본 연구를 통한 연구 기여를 다음과 같이 정리할 수 있다.

접수일: 2007년 11월 14일; 채택일: 2008년 2월 12일
* 본 연구는 한국과학재단 특정기초연구 (R01-2006-000-10609-0) 지원으로 수행되었습니다.
† 주저자, jhkimygk@gmail.com

⊙ RDF 트리플 보안 객체 기반의 권한 명세 방법의 소개 및 권한 명세의 의미 정의
⊙ 웹상의 RDF 문서에서의 세밀한 정보 은닉과 관련한 RDF 접근 제어 모델 연구
⊙ 그래프 레이블링을 통한 RDF 권한 충돌의 효율적 발견에 대한 소개

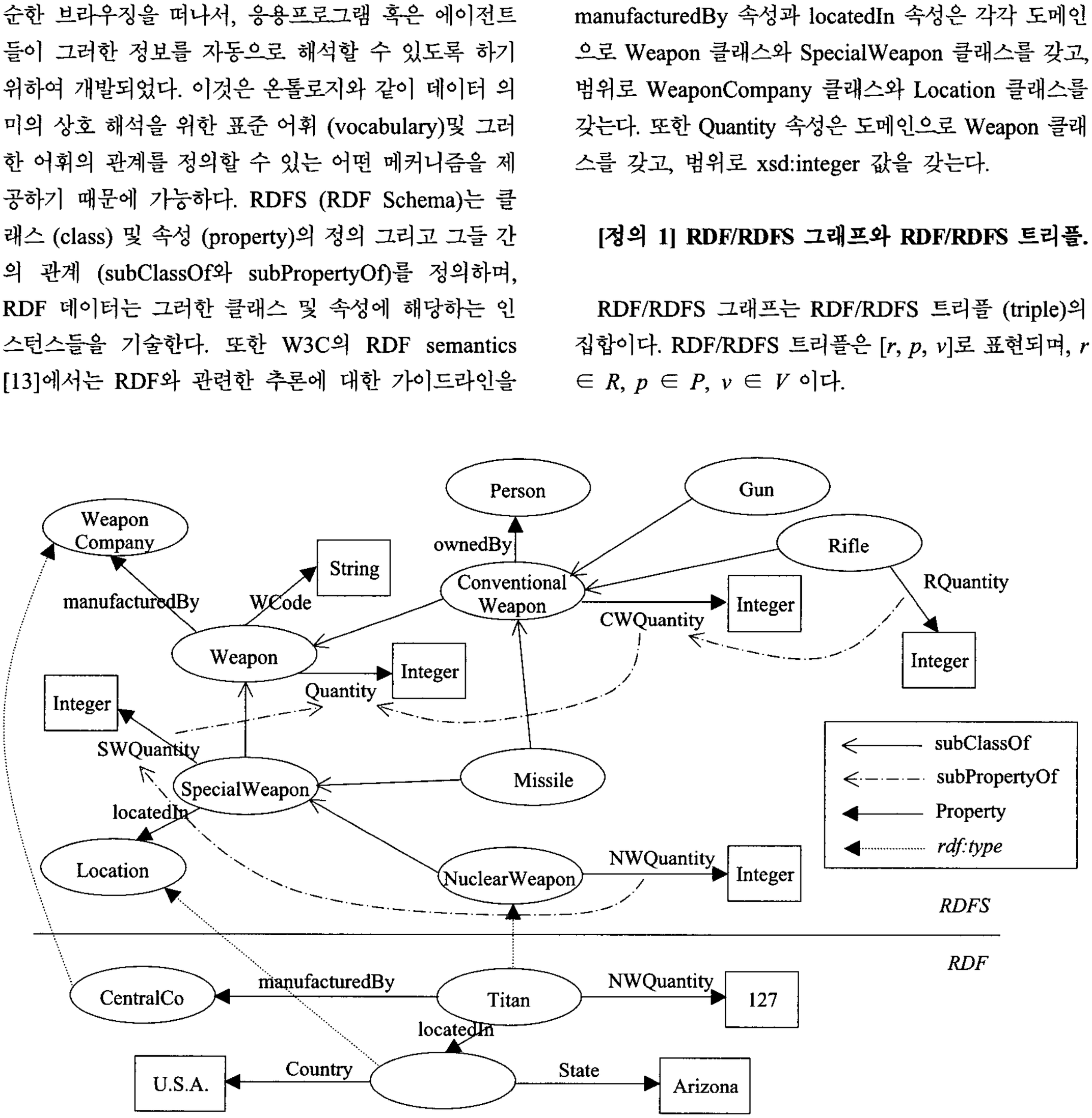
2.2. RDF에 대한 간략한 소개

RDF는 웹 사용자들의 웹 문서에 기술된 정보의 단순한 브라우징을 떠나서, 응용프로그램 혹은 에이전트들이 그러한 정보를 자동으로 해석할 수 있도록 하기 위하여 개발되었다. 이것은 온톨로지와 같이 데이터 의미의 상호 해석을 위한 표준 어휘 (vocabulary)및 그러한 어휘의 관계를 정의할 수 있는 어떤 메커니즘을 제공하기 때문에 가능하다. RDFS (RDF Schema)는 클래스 (class) 및 속성 (property)의 정의 그리고 그들 간의 관계 (subClassOf와 subPropertyOf)를 정의하며, RDF 데이터는 그러한 클래스 및 속성에 해당하는 인스턴스들을 기술한다. 또한 W3C의 RDF semantics [13]에서는 RDF와 관련한 추론에 대한 가이드라인을 제시하고 있다.

[그림 1]은 하나의 예로 부록 1의 Weapon에 관한 RDFS 문서와 [그림 2]의 RDF 문서를 그래프로 표현한 것이다 (본 예는 참고 문헌 [7]의 예를 일부 수정하여 사용한 것이다). NuclearWeapon 클래스는 SpecialWeapon 클래스의 하위 클래스이고, SpecialWeapon 클래스는 Weapon 클래스의 하위 클래스이다. 또한 NWQuantity는 SWQuantity의 하위 속성이고, SWQuantity는 Quantity의 하위 속성이다. RDFS에서 모든 속성의 정의는 범위 (range)와 도메인 (domain)을 갖게 되는데, 예로 manufacturedBy 속성과 locatedIn 속성은 각각 도메인으로 Weapon 클래스와 SpecialWeapon 클래스를 갖고, 범위로 WeaponCompany 클래스와 Location 클래스를 갖는다. 또한 Quantity 속성은 도메인으로 Weapon 클래스를 갖고, 범위로 xsd:integer 값을 갖는다.

[정의 1] RDF/RDFS 그래프와 RDF/RDFS 트리플.

RDF/RDFS 그래프는 RDF/RDFS 트리플 (triple)의 집합이다. RDF/RDFS 트리플은 $[r, p, v]$로 표현되며, $r \in R$, $p \in P$, $v \in V$ 이다.



〔그림 1〕 Weapon 관련 RDF/RDFS 그래프의 예

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:ex="http://example.org/schemas/weapons#">
   <ex:NuclearWeapon rdf:ID="Titan">
      <ex:manufacturedBy rdf:resource="ex:CentralCo"/>
      <ex:NWQuantity rdf:datatype="&xsd;integer">127</ex:NWQuantity>
      <ex:locatedIn>
         <rdf:Description><ex:Country>U.S.A</ex:Country><ex:State>Arizona</ex:State></rdf:Description>
      </ex:locatedIn>
   </ex:NuclearWeapon>
</rdf:RDF>
```

〔그림 2〕 Nuclear Weapon RDF 웹 문서의 예

- 집합 R은 RDF/RDFS에서의 클래스 혹은 인스턴스를 참조하는 URI (Uniform Resource Identifier) 노드와 공백 노드를 포함한다.
- 집합 P는 클래스의 속성 (property)을 참조하는 URI이다.
- 집합 V는 속성에 의해 관계되어지는 다른 클래스 혹은 인스턴스의 URI, 공백 노드, 그리고 리터럴 (literal) 값을 포함한다.

예로 부록 1의 RDFS 문서를 표현한 [그림 1]의 실선 윗부분의 RDFS 그래프에서는 [Weapon, manufactured By, WeaponCompany], [Special Weapon, locatedIn, Location], 그리고 [그림 2]의 RDF 웹 문서를 표현한 [그림 1]의 실선 아래 RDF 그래프에서는 [Titan, NWQuantity, 127]과 같은 RDF 트리플들이 분해될 수 있다. 또한, [그림 2]의 RDF 웹 문서에서와 같이 속성 URI 상수 p = "ex:manufacturedBy"는 r값으로 인스턴스 URI 상수 "Titan" v값으로 인스턴스 URI 상수 "ex:CentralCo"를 가지며, p "ex:NWQuantity"는 v값으로 127의 리터럴 값을 갖는다. p "ex:locatedIn"은 v 값으로 [그림 1]의 RDF 그래프에서 <rdf:Description>에 해당하는 공백 노드를 갖는다. 따라서, p "ex:Country"의 r 값은 <rdf:Description>의 공백 노드이다.

[정의 2] subClassOf와 subPropertyOf.

subClassOf는 RDFS에서 어떤 클래스 c_i가 다른 클래스 c_j의 하위 클래스임을 정의하기 위한 어휘이다. 또한 앞으로 $c_i \subset c_j$의 기호로 표현하도록 하겠다. 만약 $c_i \subset c_j$이면, c_i및 c_i의 인스턴스는 c_j의 속성을 계승한다. 예로, NuclearWeapon 또한 Weapon으로부터 계승된 manufacturedBy 속성을 갖는다. 마찬가지로, supProperty Of는 두 속성 p_i와 p_j에 대하여 $p_i \subset p_j$ 관계를 나타내며, 이러한 경우 p_i의 r과 p_j의 r은, 즉 도메인 (domain)은 같은 클래스이어야 하며, v값, 즉 범위 (range) 또한 같은 클래스이어야 한다.

[표 1]은 RDF semantics [13]의 RDFS entailment 규칙에서 제시하고 있는 RDF 데이터에 대한 몇 가지

〔표 1〕 몇 가지 주요한 RDF 추론 규칙

규칙 이름		규칙
기본 규칙	*rdfs*2	*if* [*a*, *b*, *c*] ∧ [*b*, *rdfs*:*domain*, *d*] *then* [*a*, *rdf*:*type*, *d*]
	*rdfs*3	*if* [*a*, *b*, *c*] ∧ [*b*, *rdfs*:*range*, *d*] *then* [*c*, *rdf*:*type*, *d*]
	*rdfs*5	*if* [*a*, *rdfs*:*subPropertyOf*, *b*] ∧ [*b*, *rdfs*:*subPropertyOf*, *c*] *then* [*a*, *rdfs*:*subPropertyOf*, *c*]
	*rdfs*7	*if* [*a*, *b*, *c*] ∧ [*b*, *rdfs*:*subPropertyOf*, *d*] *then* [*a*, *d*, *c*]
	*rdfs*9	*if* [*a*, *rdf*:*type*, *b*] ∧ [*b*, *rdfs*:*subClassOf*, *c*] *then* [*a*, *rdf*:*type*, *c*]
	*rdfs*11	*if* [*a*, *rdfs*:*subClassOf*, *b*] ∧ [*b*, *rdfs*:*subClassOf*, *c*] *then* [*a*, *rdfs*:*subClassOf*, *c*]
확장 규칙	*ext*1	*if* [*a*, *rdfs*:*domain*, *b*] ∧ [*b*, *rdfs*:*subClassOf*, *c*] *then* [*a*, *rdfs*:*domain*, *c*]
	*ext*2	*if* [*a*, *rdfs*:*range*, *b*] ∧ [*b*, *rdfs*:*subClassOf*, *c*] *then* [*a*, *rdfs*:*range*, *c*]
	*ext*3	*if* [*a*, *rdfs*:*domain*, *b*] ∧ [*c*, *rdfs*:*subPropertyOf*, *a*] *then* [*c*, *rdfs*:*domain*, *a*]
	*ext*4	*if* [*a*, *rdfs*:*range*, *b*] ∧ [*c*, *rdfs*:*subPropertyOf*, *a*] *then* [*c*, *rdfs*:*range*, *a*]

주요한 추론을 정리한다.

먼저, 규칙 rdfs9와 rdfs11은 subClassOf 추론과 관련된다. 예로, [그림 1]에서 [Titan, rdf:type, Nuclear Weapon] 이고 [NuclearWeapon, rdfs:subClassOf, SpecialWeapon] 그리고 [SpecialWeapon, rdfs:sub ClassOf, Weapon] 이면, rdfs11에 의해 [Nuclear Weapon, rdfs:subClassOf, Weapon]이 추론되고, rdfs9에 의해 [Titan, rdfs:type, SpecialWeapon] 그리고 [Titan, rdfs:type, Weapon] 등이 추론된다. 다음으로 규칙 rdfs5와 rdfs7은 subPropertyOf 추론과 관련된다. 예로, [NWQuantity, rdfs:subPropertyOf, SWQuantity], [SWQuantity, rdfs:subPropertyOf, Quantity], 그리고 [NuclearWeapon, NWQuantity, 리터럴 상수] 이면, rdfs5에 의해 [NWQuantity, rdfs:subPropertyOf, Quantity]가 추론되고, rdfs7에 의해 [NuclarWeapon, SWQuantity, 리털럴 상수] 그리고 [NuclearWeapon, Quantity, 리터럴 상수] 등이 추론된다. 다음으로 rdfs2, rdf3, 그리고 확장 규칙들은 범위 및 도메인 추론에 관련된다. 예로, [Titan, manufacturedBy, CentralCo], [manufacturedBy, rdfs:range, WeaponCompany], 그리고 [Weapon Company, rdfs:subClass Of, Company] 이면, 먼저 ext2에 의해 [manufacturedBy, rdfs:range, Company]가 추론되고, rdfs3에 의해 [CentralCo, rdf:type, Company]가 추론된다. 본 연구에서는 위에 설명된 RDF 추론의 핵심인 subClassOf 추론, subPropertyOf 추론, 그리고 범위 및 도메인 추론에 초점을 맞추어 RDF 접근 권한의 충돌 문제를 살펴 볼 것이다.

Ⅱ. 관련 연구

현재 시맨틱 웹과 관련한 보안 연구는 미미한 실정이다. 본 절에서는 RDF 데이터에 대한 몇 가지 주요한 접근 제어 관련 연구를 소개하며, 본 연구와의 차이점을 간단히 언급하도록 하겠다.

먼저 Damiani et al [3]과 E. Bertino et al. [4, 5]은 XML 문서에 대한 세밀한 (fine-grained) 접근 제어 모델을 제시하였다. 명세된 접근 제어에 따라, XML 트리 구조에서의 각 노드를 사용자에게 보이지 않도록 (invisible) 한다. 비록 RDF 문서가 XML로 기술되지만, 제안된 XML 접근제어 모델은 서론에서 언급한 것처럼 중요한 제약 사항을 갖는다. RDF는 소규모의 온톨로지 데이터로 간주될 수 있으므로, 추론 (혹은 RDF Entailment [13]라고 불려짐)이 존재하고, 따라서 추론에 의해 새롭게 생성된 정보에 대한 접근 제어를 고려할 수 있어야 한다.

Jain과 Farkas [6]는 RDF Entailment를 고려한 새로운 RDF 접근 제어 모델을 제시하였다. 하지만 Jain과 Farkas는 그들의 연구에서 RDF 트리플 기반의 접근 권한을 보안 관리자가 어떻게 명세하며, 명세의 의미가 어떻게 해석될 수 있는 지를 설명하고 있지 않다. 단지 어떤 명세에 의해서 이미 할당된 접근권한들을 가정하고, 이에 대한 추론과 관련한 권한 충돌 문제를 다룬다. 하지만, 결국 권한의 충돌은 권한 명세와 권한 추론 사이의 충돌이기 때문에, 권한 명세의 방법과 의미를 먼저 정의하는 것은 중요한 문제이다. 또한 권한 충돌의 발견을 위한 알고리즘으로 추론에 의해 RDF 인스터스에 새롭게 할당된 접근권한을 모두 조사하는 매우 단순 소모적인 (brute force) 방법을 사용한다. 본 연구에서는 권한 명세에 의한 권한 전파와 추론에 의한 권한 전파 사이의 충돌 유형을 파악함으로서 권한 충돌을 효율적으로 발견하며, 이를 위해 RDF 포함 (subsumption) 관계에 기반한 그래프 레이블링 기법을 사용한다.

Qin과 Atluri [7]는 온톨로지 개념들 사이의 다양한 관계에 있어서의 추론에 의한 권한 전파 문제 및 충돌 문제를 소개한다. 즉, subClassOf, subPropertyOf 외에 두 개념 사이의 동등 관계 (equivalence relationship), 전체 개념과 부분 개념 사이의 관계 (예로, OWL [2]에서의 intersectionOf, unionOf), 비 추론적 관계 (non-inferable relationship) (예로, OWL에서의 disjointWith와 intersectionOf)등 다양한 의미적 관계에 대한 권한 전파를 고려한다. 하지만 제안된 접근 제어 모델의 보안 객체는 XPath와 같은 RDF path이며, RDF 트리플 구조에 기반을 두지 않는다. 따라서 RDF 트리플을 기본 구조로 하는 RDF와 OWL 모델 [1, 2]의 의미, 더더욱 RDF entailment [13]에 직접적으로 부합되지는 않는다. 즉, 접근 권한 명세 및 추론에 의한 접근 권한 전파에 있어 본 연구과 다른 의미를 갖는다. 또한 권한 충돌의 문제에 있어서도 Qin과 Atluri는 그들이 정의한 추론에 의한 권한 전파 규칙들 사이에서의 충돌 문제만을 살펴본다. 하지만 본 연구에서는 권한 명세시의 하위 개념으로의 권한 전파와 추로시의 상위 개념으로의 권한 전파 사이에서의 권한 충돌 문제를 살펴본다. 본 논문에서 제안하는 RDF 트리플 기반의 접근 제어 모델을 OWL 데이터에 적용할 수 있도록 확장하는 것은 본 연구의 향

후 연구 과제이다.

Javanmardi et al. [8] 또한 온톨로지의 다양한 관계에서의 추론에 의한 권한 전파 문제 및 충돌 문제를 소개한다. 하지만 그들 역시 RDF 트리플 구조에 기반하지 않으며, 상/하위 노드로의 두 가지 권한 전파에 의한 권한 충돌 문제를 다루는 것은 아니다.

Kaushik et al. [9]은 본 연구에서의 RDF 웹 문서의 세밀한 정보 은닉의 응용처럼, 사용자에 따라 웹 온톨로지 데이터의 특정 부분을 보이거나 보이지 않게 하기 위한 접근 제어 모델을 제시하였다. 하지만, 그들 연구의 주안점은 RDF 데이터 정보 은닉에 대한 형식적 틀(formal framework)을 처음으로 소개하는 것이며, 또한 정보 은닉의 다양한 방법을 RDF 데이터에 적용한 것이다. 즉, 사용자에 따라, RDF 데이터의 특정 서브 그래프가 접근 불허되거나, 혹은 어떤 클래스 혹은 속성 노드만이 접근 허용되거나, 또는 접근 불허 클래스 혹은 속성의 이름을 다른 이름으로 변경하여 보이게 하는 등의 다양한 정보 은닉 방법을 소개한다. 하지만 그들은 본 연구에서의 주안점인 명시적인 접근 불허의 데이터뿐만 아니라 불허된 추론을 일으킬 수 있는 데이터 또한 보이지 않게 하기 위한 추론에 의한 권한 충돌 문제를 다루지 않는다.

Reddivari et al. [10]은 RDF 데이터에 대한 접근권한에 있어 보안 객체에 대하여 수행될 수 있는 다양한 RDF 연산, 예로 삽입 (insert, insertModel, insertSet), 삭제 (remove, removeModel, removeSet), 갱신 (update), 읽기 (see, use)등을 소개하고 있다. 본 연구에서는 웹 사용자의 RDF 웹 문서에 대한 브라우징에 대한 접근 제어를 고려하므로 단지 읽기 연산만을 고려한다.

Ⅲ. RDF 접근 권한 명세의 정의

3.1. RDF 보안 객체

먼저 접근권한 명세에서의 보안 객체 (security object)가 되는 RDF 패턴에 대하여 설명한다. RDF 패턴은 RDF 트리플 구조를 따르므로 보안 관리자로 하여금 접근 권한 객체를 쉽고 명료하게 명세할 수 있도록 한다. RDF 패턴은 여러 개의 RDF 트리플들을 하나의 패턴으로 묶을 수 있다.

[정의 3] RDF 패턴.

RDF 패턴 (pattern)은 RDF 트리플 구조의 [*r*, *p*, *v*] 형태를 가진다. *r*, *p*, *v*는 임의의 변수 $*x*, $*y*, $*z* 등으로 치환될 수 있으며, 본 연구에서 *v* 값은 항상 임의의 변수 $*z*를 가짐을 정의한다. 즉, *v* 값에 따른 더욱 세밀한 (fine-grained) 접근 제어를 고려하지 않는다. 또한 정의 1에서의 *r*, *v*의 값이 공백 노드를 갖는 경우를 제외하기로 한다.

예로, 패턴 t_i = [$*x*, NWQuantity, $*z*]는 [그림 1]의 RDF/RDFS 그래프 표현에서의 $\mu(t_i)$ = {[Nuclear Weapon, NWQuantity, 리터럴 상수], [Titan, NWQuantity, 리터럴 상수]}와 매치된다. 또한 매칭 트리플 μ([NuclearWeapon, $*y*, $*z*]) = {[NuclearWeapon, manufactureBy, WeaponCompa ny], [NuclearWeapon, WCode, 리터럴 상수], [NuclearWeapon, Quantity, 리터럴 상수], [NuclearWeapon, SWQuantity, 리터럴 상수], [NuclearWeapon, locatedIn, Location], [Nuclear Weapon, NWQuantity, 리터럴 상수]}이며, 매칭 트리플 μ([$*x*, $*y*, $*z*])는 그래프의 모든 에지를 매치한다.

3.2. 접근 권한

다음으로 접근 권한은 다음과 같이 명세된다.

[정의 4] RDF 접근 권한.

RDF 접근권한 *a*는 <*subject*, *object*, *action*, *sign*, *type*>의 다섯 개의 튜플로 구성된다.

- *subject*는 권한이 주어지는 주체이다.
- *object*는 권한이 적용되는 객체로 정의 3의 RDF 패턴에 의하여 매칭되는 RDF 트리플들이다.
- *action*은 *object*에 대해 수행되는 연산을 기술하며, 본 연구에서는 웹 문서 상에서의 RDF 데이터의 접근 제어를 고려하므로, 읽기 연산으로 제한한다.
- *sign* ∈ {+, -}이며, + 부호에 의해 *action*이 허용됨을 - 부호에 의해 *action*이 거부됨을 표시한다.
- *type* ∈ {L, R}이며, L (Local)은 권한의 전파(propagation)가 해당 RDF 트리플에만 적용되는 것을 표시하며, R (Recursive)는 권한의 전파가 해당 RDF 트리플의 모든 하위 RDF 트리플 (이것은 subClassOf, subPropertyOf의 포함 관계에 의하여 정의되어짐)에 적용됨을 표시한다.

3.3. 접근 권한에 의한 RDF/RDFS 웹 문서의 정보 은닉

본 연구에서의 RDF 접근 제어의 응용은 웹 사용자에 따른 RDF 웹 문서의 세밀한 (fine-grained) 정보 은닉 (information hiding)이다. 예로, <Dave, [$x, manufactured By, $z], read, -, R> 접근 권한에 따라, 웹 사용자 Dave에게는 [그림 2]의 RDF 문서에서 <manufactured By> 속성 부분이 보이지 않아야 한다. 본 절에서는 명세된 접근 권한에 따라 RDF 데이터의 어떤 부분이 은닉되어야 하는 지를 정의한다. 먼저 RDF 문서에서의 정보 은닉의 부분은 RDF 트리플에서의 *v* 값의 유형 ∈ {클래스 혹은 인스턴스 URI 상수, 공백 노드, 리터럴 상수}에 따라 달리 결정되며, [표 2]는 이를 정리한다. 또한 RDFS에서의 정보 은닉 부분은 RDF 패턴에서의 *r*, *p* 값에 따라 해당 클래스 혹은 속성의 정의 부분이 보이지 않게 된다. 예 1과 2를 통하여 이를 설명한다.

예 1. 접근 권한 <Dave, [$x, manufacturedBy, $z], read, -, R>에 따라, Dave는 [그림 2]의 RDF 문서에서의 <ex:manufacturedBy rdf:resource= "ex:Central Co"/> 부분과 부록 1의 RDFS 문서에서의 manufacturedBy 정의 부분, 즉 <rdf:Description rdf:ID = "manufacturedBy"> ... </rdf:Description> 부분을 볼 수 없다. 하지만, 이 경우 URI "ex:CentralCo"가 참조하는 실제 개체는 은닉되지 않는다. 단지 URI의 참조 관계만을 볼 수 없게 만드는 것이다. 다음으로 접근 권한 <Dave, [SpecialWeapon, locatedIn, $z], read, -, R>을 가정하자. [그림 2]의 RDF 문서에서의 해당 *v* 값이 "ex:locatedIn"의 <rdf:Description>과 같이 공백 노드이기 때문에, RDF 문서로부터 속성 locatedIn 및 공백 노드를 포함하는 하위 트리 (subtree) 전체가 보이지 않게 된다. 즉, <ex:locatedIn> <rdf:Description><ex:Country>U.S.A</ex:Country><ex:State>Arizona</ex:State></rdf:Description></ex:locatedIn> 부분 전체를 볼 수 없다. 다음으로 RDFS 문서에서 <rdf:Description rdf:ID = "locatedIn"> ... </rdf:Description> 정의 부분이 보이지 않게 된다. 다음으로 <Dave, [Nuclear Weapon, NWQuantity, $z], read, -, R>을 가정하자. [그림 2]의 RDF 문서에서 해당 *v* 값이 리터럴이기 때문에 <ex:NWQuantity rdf:datatype="&xsd;integer"> 127 </ex:NWQuantity> 부분은 보이지 않아야 한다. 또한 RDFS 문서에서의 속성 NWQuantity에 관한 정의 부분은 보이지 않아야 한다.

예 2. 접근 권한 <Dave, [NuclearWeapon, $y, $z], read, -, L>에 따라, Dave는 NuclearWeapon의 모든 속성을 볼 수 없다. 만약 어떤 클래스의 모든 속성을 볼 수 없으면, 그 클래스 전체를 보이지 않게 한다. 따라서 [그림 2]의 RDF 문서에서 <ex:NuclearWeapon rdf:ID ="Titan"> ... </ex: NuclearWeapon> 전체가 보이지 않게 된다. 또한 부록 1의 RDFS 문서에서 Nuclear Weapon과 관련된 클래스 및 모든 속성 정의 부분, 즉 <rdf:Description rdf:ID = "NuclearWeapon">...</rdf:Description> 부분과 <rdf:Description rdf:ID = "NW Quantity">...</rdf:Description> 부분이 보이지 않게 된다.

〔표 2〕 접근 권한 명세에 따른 RDF 웹 문서의 정보 은닉 부분

	read (-)
URI 상수	- 어떤 클래스 혹은 인스턴스 *r*로부터 속성 *p*가 보이지 않아야 함. - 또한 *v* 값인 URI 상수 값 또한 보이지 않아야 함. 하지만, 이것이 URI가 참조하는 개체를 보이지 않게 하여야 하는 것을 의미하지는 않음. 단지, URI 값만을 보이지 않게 함.
공백 노드 (blank node)	- 어떤 개념 혹은 인스턴스 *r*로부터 속성 *p*가 보이지 않아야 함. - 또한 속성 *p*의 값인 공백 노드 또한 보이지 않아야 함.
리터럴 상수 (literals)	- 어떤 개념 혹은 인스턴스 *r*로부터 속성 *p*가 보이지 않아야 함. - 또한 속성 *p*의 값인 리터럴 값 또한 보이지 않아야 함.

Ⅳ. 접근 권한 명세에서의 하위 개념으로의 권한 전파

접근 권한 명세는 *type*이 R일 경우 하위 클래스 혹은 하위 속성으로의 권한 전파가 이루어진다. 본 절에서는 추론에 의한 권한 충돌을 소개하기에 앞서, 접근 권한 명세에서의 하위 클래스 혹은 하위 속성으로의 권한 전파가 어떻게 이루어지는 지를 정의한다. 특별히 본 연구에서의 권한 전파의 의미는 비록 어떤 권한의 명세가 그것에 의해 전파된 권한들의 의미를 함께 내포하지만, 이것이 하위 클래스 혹은 하위 속성으로의 전파된 권한들을 접근 권한 저장소에 별도로 추가 생성해야하는 것을 의미하지 않음을 언급해 둔다.

4.1. subClassOf와 관련한 권한 전파

정의 2에 따라 어떤 클래스 c_i가 c_j의 하위 클래스 일 때, c_i는 c_j의 속성을 계승한다. 따라서 c_j의 속성 p_k에 대한 접근 권한 ca_j는 c_i에서도 그대로 적용된다. 이러한 subClassOf 관계에 의한 권한 전파의 의미를 접근 허용(+)의 경우 ca_j+ → ca_i+, 접근 불허(-) 의 경우 ca_j- → ca_i-로 표현하기로 한다 (본 논문에서의 권한 전파와 관련한 표현 기호 및 개념은 Qin et al. [7]에서 사용한 표현 형식을 일부 활용한 것임을 밝힌다).

예 3. 만약 접근 권한 <Dave, [SpecialWeapon, SW Quantity, $z], read, -, R>이 명세 될 경우, 이것은 sub ClassOf 관계에 의해 <Dave, [NuclearWeapon, SW Quantity, $z], read, -, R>과 <Dave, [Missile, NW Quantity, $z], read, -, R>의 접근 권한을 함께 내포한다. 앞서 언급하였듯이, 전파된 접근 권한은 추가적으로 생성되지는 않으며, 단지 접근 권한 저장소에는 <Dave, [SpecialWeapon, SWQuantity, $z], read, -, R>이 존재한다. +의 경우도 같은 방식으로 전파된다.

예 4. 접근 권한 <Dave, [SpecialWeapon, $y, $z], read, -, R>의 경우는 SpecialWeapon의 모든 속성에 대한 접근 권한이 하위 클래스에 계승되며, 따라서 계승된 각각의 속성에 대한 접근 권한이 하위 클래스에도 동일하게 적용됨을 의미한다. 즉, 접근권한 <Dave, [Nuclear Weapon, SWQuantity, $z], read, -, R>, <Dave, [Missile, SWQuantity, $z], read, -, R>, <Dave, [Nuclear Weapon, locatedIn, $z], read, -, R>, <Dave, [Missile, locatedIn, $z], read, -, R>의 의미를 동시에 내포한다.

예 5. 접근권한 <Dave, [SpecialWeapon, *, *], read, -, R>의 경우는 SpecialWeapon의 모든 속성에 대한 접근 권한뿐만 아니라, 하위 클래스의 모든 속성에 대한 접근을 불허함을 의미한다. 즉, <Dave, [SpecialWeapon, $y, $z], read, -, R>, <Dave, [NuclearWeapon, $y, $z], read, - R>, <Dave, [Missile, $y, $z], read, -, R>의 의미를 함께 내포한다.

[정의 5] RDF * 패턴.

이것은 예 5에서와 같이 RDF 패턴이 [*r*, *, *]로 표현되며, *r*의 모든 하위 클래스에서의 *r*로부터 계승된 모든 속성뿐만 아니라, 하위 클래스 자신의 모든 속성을 한 번에 매칭하기 위한 특별한 RDF 패턴이다.

4.2. subPropertyOf와 관련한 권한 전파

어떤 속성 p_i가 p_j의 하위 속성일 때, 속성 p_j에 대한 접근 권한 pa_j는 p_i에서도 그대로 적용된다. 이러한 subPropertyOf 관계에 의한 권한 전파의 의미를 접근 허용(+)의 경우 pa_j+ → pa_i+, 접근 불허(-)의 경우 pa_j- → pa_i-로 표현하기로 한다.

예 6. 만약 <Dave, [ConventionalWeapon, CW Quantity, $z], read, -, R>이 명세 될 경우, 이것은 subPropertyOf 관계에 의해 <Dave, [Rifle, RQuantity, $z], read, -, R>의 접근 권한 또한 함께 내포한다.

예 7. 접근 권한 <Dave, [Weapon, $y, $z], read, -, R>의 경우, Weapon의 모든 속성들의 하위 속성들에도 동일한 접근권한이 적용된다. 마찬가지로 RDF * 패턴의 경우, 즉 <Dave, [Weapon, *, *], read, -, R> 경우, Weapon의 하위 클래스들의 모든 속성들의 하위 속성들에도 동일한 접근 권한이 적용된다.

4.3. rdf:type과 관련한 권한 전파

어떤 클래스 c_i에 대한 접근 권한 ca_i가 명세 될 경우, c_i의 인스턴스들도 ca_i를 따른다. 마찬가지로, 어떤 속성 p_k에 대한 접근 권한 pa_k가 명세 될 경우, p_k를 갖는 인스턴스에도 동일한 접근 권한이 적용된다.

V. RDF 추론에서의 상위 개념으로의 권한 전파 및 권한 충돌 유형

본 절에서는 접근 권한이 명세 될 경우, RDF 추론에 의한 접근 권한 전파에 따른 권한 충돌 문제를 살펴본다. 본 연구에서는 RDF 추론의 핵심이 되는 포함 관계 기반 추론 (즉, subClassOf와 subPropertyOf 추론)과 범위 및 도메인 추론에 초점을 맞춘다.

5.1. subClassOf의 추론에 의한 권한 전파

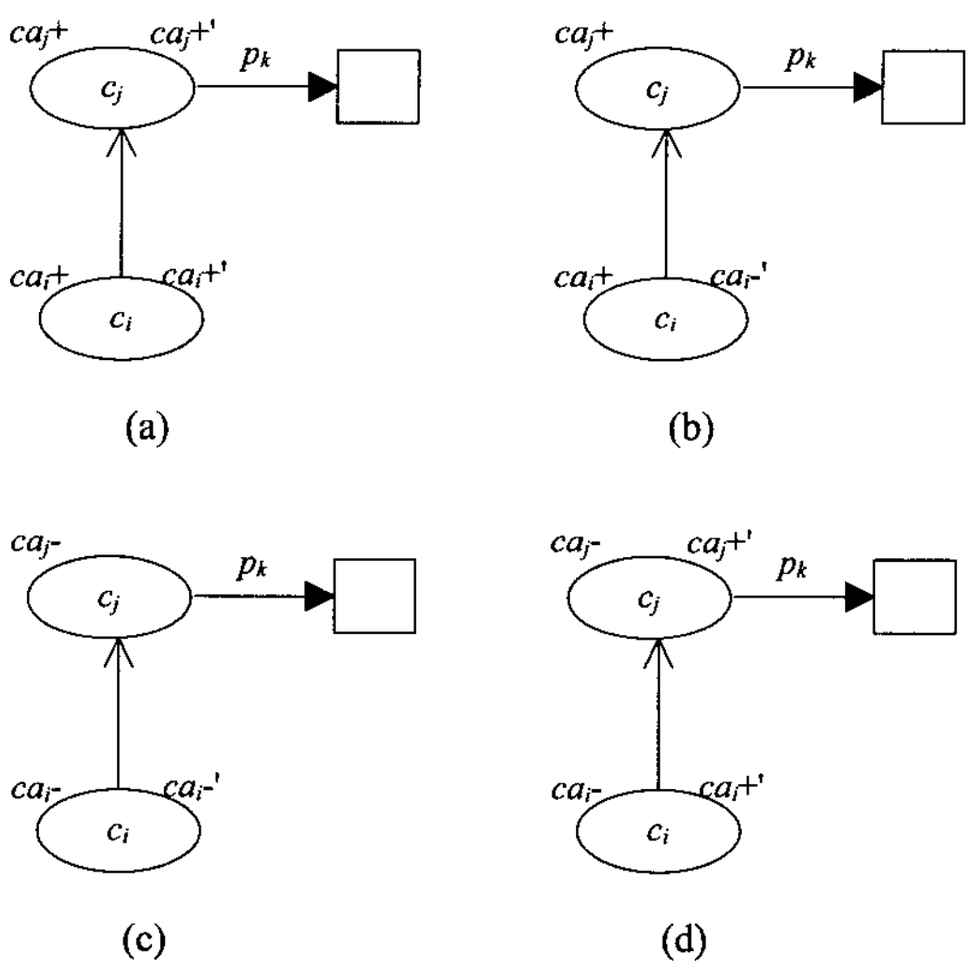
⊙ $c_i \subset c_j$, ca_j+ → ca_i+, ca_i+' => ca_j+': [그림 3(a)]에서 c_i가 c_j의 하위 클래스이고 접근 권한 명세에 의한 ca_j+ → ca_i+ 권한 전파를 고려하자. 또한, 이후 [c_i, p_k, $z]에 새롭게 명세된 ca_i+'의 접근 권한을 고려하자. subClassOf 추론에 의하여 RDF 트리플 t_i=[c_i, p_k, $z]로부터 t_j=[c_j, p_k, $z]를 해석

할 수 있으므로, 추론에 의한 권한 전파 ca_i+' => ca_j+'를 고려할 수 있다. 이 경우 $sign(ca_j+) \equiv sign(ca_j+')$ 이므로 충돌이 발생하지 않는다. 예로, ca_j+ = <Dave, [SpecialWeapon, locatedIn, $z], read, +, R>이 명세될 경우 권한 전파에 의하여 ca_i+ = <Dave, [NuclearWeapon, locatedIn, $z], read, +, R>을 포함한다. 이후 ca_i+' = <Dave, [NuclearWeapn, locatedIn, $z], +, R>이 새롭게 명세될 경우 [표 1]의 추론 규칙 rdfs9와 rdfs11에 의하여 [NuclearWeapon, locatedIn, $z]로부터 [Special Weapon, locatedIn, $z]가 추론 될 수 있으므로, 추론에 의한 권한 전파 ca_j+' = <Dave, [SpecialWeapon, locatedIn, $z], read, +, L>을 고려할 수 있다. 이 경우 $sign(ca_j+) \equiv sign(ca_j+')$ 이므로 충돌이 발생하지 않는다.

⊙ $c_i \subset c_j$, $ca_j+ \rightarrow ca_i+$, $ca_i-' \not\Rightarrow ca_j-'$: 마찬가지로, [그림 3(b)]의 $ca_j+ \rightarrow ca_i+$ 권한 전파를 고려하자. 또한 이후 [c_i, p_k, $z]에 별도로 명세된 ca_i-'의 접근권한을 고려하자. 이 경우 ca_i-'이므로, 클래스 c_i의 속성 p_k에 대한 접근이 허용되지 않게 되므로 c_j로의 subClassOf 추론은 수행될 수 없다. 따라서 추론에 의한 권한 전파 충돌은 발생할 수 없다. 예로, ca_i-' = <Dave, [NuclearWeapn, locatedIn, $z], -, R> 의 경우, 부록 1과 [그림 2]의 RDF/RDFS 문서에서 속성 locatedIn과 관련한 부분은 Dave에게 보여 지지 않는다. 따라서 locatedIn과 관련한 subClassOf 추론은 당연히 수행될 수가 없다.

⊙ $c_i \subset c_j$, $ca_j- \rightarrow ca_i-$, $ca_i-' \not\Rightarrow ca_j-'$: [그림 3(c)]의 $ca_j- \rightarrow ca_i-$ 권한 전파를 고려하자. 또한 이후 [c_i, p_k, $z]에 별도로 명세된 ca_i-'의 접근권한을 고려하자. 이 경우 역시 ca_i-'이므로, 클래스 c_i의 속성 p_k에 대한 접근이 허용되지 않게 되므로 c_j로의 subClassOf 추론은 수행될 수 없다. 따라서 추론에 의한 권한 전파 충돌은 발생할 수 없다.

⊙ $c_i \subset c_j$, $ca_j- \rightarrow ca_i-$, $ca_i+' \Rightarrow ca_j+'$: [그림 3(d)]의 $ca_j- \rightarrow ca_i-$ 권한 전파를 고려하자. 또한, 이후 [c_i, p_k, $z]에 새롭게 명세된 ca_i+'의 접근 권한을 고려하자. subClassOf 추론에 의한 권한 전파 $ca_i+' \Rightarrow ca_j+'$를 고려할 수 있으므로, $sign(ca_j-) \neq sign(ca_j+')$ 의 충돌이 발생한다. 따라서 이 경우, ca_i+'는 허용되어서는 안 된다. 마찬가지로, $ca_j+ \rightarrow ca_i+$가 설정되어 있고 이후 *type* 값이 L인 ca_j-

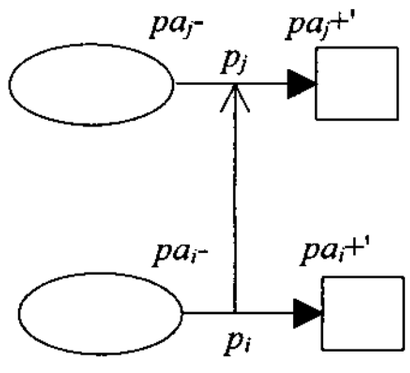


〔그림 3〕 RDF 추론에서의 권한 전파 충돌의 유형

접근 권한을 다시 설정할 경우, 충돌을 발생시키므로 이것 또한 허용되어서는 안 된다. 예로, ca_j- = <Dave, [SpecialWeapon, locatedIn, $z], read, -, R>과 이후 명세된 ca_i+' = <Dave, [Nuclear Weapn, locatedIn, $z], +, R>을 가정하자. ca_i+'에 따라 Dave에게는 RDF/RDFS 문서에서의 속성 locatedIn의 정의 부분과 데이터가 보이게 된다. 따라서 추론 규칙 rdfs9와 rdfs11에 의하여 Titan은 SpecialWeapon으로 해석될 수 있으므로, ca_i+'는 ca_j-와 충돌 관계이다.

5.2. subPropertyOf의 추론에 의한 권한 전파

subClassOf와 마찬가지로, [그림 4]와 같이 $p_i \subset p_j$, $pa_j- \rightarrow pa_i-$, $pa_i+' \Rightarrow pa_j+'$의 경우 충돌이 발생하며, 따라서 이러한 권한 설정은 허용되어서는 안 된다. 예로, pa_j- = <Dave, [SpecialWeapon, SWQuantity, $z], read, -, R>이 선언된 후, pa_i+ = <Dave, [Nuclear

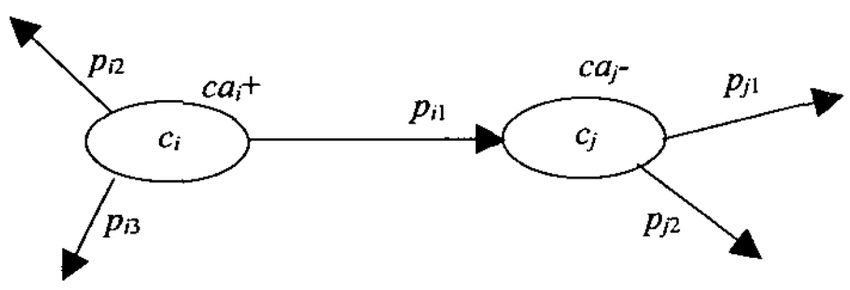


〔그림 4〕 subPropertyOf 추론에서의 권한 전파 충돌

Weapon, NWQuantity, $z], read, +, R>이 선언될 경우, 추론 규칙 rdfs5, rdfs7, rdfs11에 의하여 [Nuclear Weapon, NWQuantity, $z]는 [SpecialWeapon, SW Quantity, $z]으로 해석될 수 있으므로 충돌이다.

5.3. 범위 및 도메인 추론에 의한 권한 전파

범위 및 도메인 추론에 있어서는 subClassOf와 subPropertyOf에서와 같은 상위 클래스 혹은 상위 속성으로의 권한 전파는 이루어 지지 않는다. 하지만 범위 추론에서 다음과 같은 충돌 가능성의 경우가 존재할 수 있다. [그림 5]의 RDF 트리플 t_i=[c_i, p_{i1}, c_j]에 대한 접근 권한 ca_i+를 고려하자. 또한 t_j=[c_j, $y, $z]에 대한 모든 속성에 대한 접근 불허를 선언하는 접근 권한 ca_j-를 고려하자; 즉, $p(t_j)$ = $y이므로 예 2에서 정의한 것처럼, c_j 전체가 접근 불허 되어야 한다. 하지만, 범위 추론에 의해 t_i의 $v(c_j)$로부터 c_j를 추론할 수 있으므로, 보안 위반 (security violation)이 일어날 수 있다. 예로, ca_i+ = <Dave, [Titan, manufacturedBy, $z], read, +, L>와 ca_j- = <Dave, [WeaponCompany, $y, $z], read, -, L>을 가정하자. 그러면, ca_j-에 의해 부록 1의 RDFS 문서에서 WeaponCompany와 관련된 클래스 정의 부분과 모든 속성 정의 부분은 Dave에게 보이지 않게 될 것이다. 하지만, ca_i+에 의해 [그림 2]의 RDF 문서에서 <ex:manufacturedBy rdf:resource="ex:CentralCo"/>가 Dave에게 보이고, 또한 부록 1의 RDFS 문서에서 manufacturedBy 속성에 대한 정의가 Dave에게 보이게 되므로, manufacturedBy 속성의 범위 추론에 의하여 CentralCo는 NuclearWeapon의 Weapon Company라는 정보를 추론할 수 있다. 따라서 p() = $y인 ca_j-의 r 값을 범위로 가지는 ca_i+는 허용되어서는 안 된다. 하지만 p() ≠ $y인 경우, 즉 어떤 특정 속성 p_{j1}에 대한 접근 불허를 선언하는 접근 권한의 경우 <Dave, [c_j, p_{i1}, $z], read, -, L>, c_j 전체를 보이지 않게 하는 것은 아니므로 충돌이 아니다.



〔그림 5〕 범위 (range) 추론에서의 권한 전파 충돌

도메인 추론과 관련하여서는, 정의 3, 4에 의한 접근 권한의 명세가 [r, p, v]에서의 r 값을 도메인으로 갖는 속성 p에 대한 접근 권한을 기술하는 것이므로, 범위 추론에서와 같은 보안 위반 현상은 일어나지 않는다.

Ⅵ. subClassOf와 subPropertyOf 추론에서의 그래프 레이블링을 통한 권한 충돌의 효율적 발견

Jain과 Farkas [6]는 접근 권한들 사이에서의 추론에 의한 충돌을 발견하기 위하여 각 접근 권한을 모든 인스턴스에 적용하고 추론한 후 충돌이 발생하는지를 모든 인스턴스에 대하여 다시 조사한다. 하지만, 앞 장에서 소개한 subClassOf와 subPropertyOf 추론에 있어서는 조상/후손 (ancestor/descendant) 관계에 있는 클래스 혹은 속성에 관한 접근 권한만을 검증함으로서, 효율적으로 접근 권한의 충돌을 발견할 수 있다. 본 장에서는 그래프 레이블링 기법 [11, 12]을 이용하여 효율적으로 권한 충돌을 발견하는 방법을 간단히 소개한다. 그래프 레이블링 기법으로는 프리픽스 (prefix) 기법 [12], 인터벌 (interval) 기법 [12], 소수 (prime number)기법 [11]등이 연구되었으며, 본 논문에서는 이의 구체적인 설명은 생략하기로 한다. 단지, 그래프 레이블링에 의한 조상/후손 관계의 정보를 이용하여 어떻게 충돌 가능성의 접근 권한을 효율적으로 발견하는지를 설명한다. 제안하는 방법의 핵심 아이디어는 Ⅴ장에서의 "$c_i \subset c_j$, ca_j- → ca_i-, ca_i+' => ca_j+'" 관찰에 의하여, +의 접근권한이 설정될 경우 조상 관계에 있는 클래스 혹은 속성에 대한 - 접근 권한의 존재 여부를 조사하는 것이고, 반대로 *type* L의 -의 접근 권한이 설정될 경우 후손 관계에 있는 클래스 혹은 속성에 대한 + 접근 권한의 존재 여부를 조사하는 것이다. 이러한 조상/후손 관계의 신속한 파악을 위하여 그래프 레이블링 기법이 이용된다.

제안하는 방법의 두 번째 핵심 아이디어는 정의 3과 5의 RDF 패턴, 즉 접근 권한 보안 객체에서의 p 값의 유형 ∈ {$y, URI 상수, *}에 따른 조상/후손 관계의 클래스 혹은 속성에서의 선별적 충돌 판정을 수행하는 것이다. 이를 [그림 6]의 표로 정리하였으며, [그림 7]의 접근 권한의 예들을 통하여 설명하도록 한다.

접근 권한 R4 = <Dave, [NuclearWeapon, $y, $z], read, +, L>이 명세될 경우, NuclearWeapon의 조상 클래스와 관련한 접근 권한은 R1이다. p(R4) = $y, p(R1) = $y, *sign*(R4) = +, *sign*(R1) = - 이므로, 이것은 [그림

조상(-) / 후손(+)	$y	URI 상수	*
$y	1)충돌	2)충돌	3)충돌
URI 상수	4)조사	5)조사	6)충돌
*	7)충돌	8)충돌	9)충돌

(a) subClassOf 관계

조상(-) / 후손(+)	$y	URI 상수	*
$y	1)충돌	2)충돌	3)충돌
URI 상수	4)충돌	5)충돌	6)충돌
*	7)충돌	8)충돌	9)충돌

(b) subPropertyOf 관계

〔그림 6〕 조상(+)/후손(-) 관계에서의 속성의 유형에 따른 권한 충돌 판정

6(a)] 표의 1)에 의하여 무조건 충돌이다. 왜냐하면 p(R4) = $y는 SpecialWeapon으로 부터의 계승되는 모든 속성을 포함하기 때문이다. 다시 접근 권한 R4 = <Dave, [Weapon, manufacturedBy, $z], read, -, L>이 명시된다고 하자. Weapon의 후손 클래스와 관련한 접근 권한은 R1, R2, R3 모두인데, R2의 경우 p(R4) = manufacturedBy, p(R2) = $y, *sign*(R4) = -, *sign*(R2) = + 이므로 [그림 6(a)] 표의 2)에 의하여 무조건 충돌이다. 왜냐하면 p(R2) = $y는 Weapon으로부터 계승되는 manufacturedBy 속성을 포함하기 때문이다. 다시 접근 권한 R4 = <Dave, [Weapon, *, *], read, -, R>를 고려하자. 역시 Weapon의 후손 클래스와 관련한 접근 권한은 R1, R2, R3인데, R2의 경우 p(R4) = *, p(R2) = $y, *sign*(R4) = -, *sign*(R2) = + 이므로 [그림 6(a)] 표의 3)에 의하여 무조건 충돌이다. 왜냐하면 p(R2) = $y는 Weapon으로 부터의 계승되는 모든 속성을 포함하기 때문이다. 다시 접근 권한 R4 = <Dave, [Nuclear Weapon, locatedIn, $z], read, +, L>를 고려하자. p(R4) = locatedIn, p(R1) = $y, *sign*(R4) = +, *sign*(R1) = - 이므로, 이것은 [그림 6(a)] 표의 4)에 의하여 충돌 가능하다. p(R4) = locatedIn은 SpecialWeapon으로부터 계승되기 때문에 최종 판정은 충돌이다. 하지만, [그림 6(a)] 표의 4)의 경우 p 값 즉 URI 상수가 조상 클래스로부터 계승되지 않는 속성이면 충돌이 발생하지 않게 되므로, URI 상수가 계승되는지 여부를 조사할 필요가 있다. [그림 6(a)] 표의 5)의 경우도 두 URI 상수 값이 같은지 여부를 조사할 필요가 있다. 6), 7), 8), 9)의

```
R1:  <Dave, [SpecialWeapon, $y, $z], read, -, R>
R2:  <Dave, [Rifle, $y, $z], read, +, L>
R3:  <Dave, [ConventionalWeapon, CWQuantity, $z],
     read, +, R>
```

〔그림 7〕 접근 권한 저장소에 저장되어 있는 권한의 예

경우는 위의 예들과 동일하다.

[그림 6(b)]의 subPropertyOf 관계에서는 모든 경우에 대하여 충돌이다. 이것은 왜냐하면 subClassOf 관계에서는 클래스의 계승 관계를 정의하기 때문에 [그림 6(a)]의 4)와 5)의 경우처럼 계승 관계에 있지 않은 속성이 존재할 수 있지만, subPropertyOf는 직접적으로 속성 사이의 계승 관계를 정의하기 때문이다. 예로 접근 권한 R4 = <Dave, [Weapon, $y, $z], read, -, R> 혹은 <Dave, [Weapon, Quantity, $z], read, -, R>이 명시될 경우, 후손 관계의 속성에 대한 접근 제어 R3와 무조건 충돌이다. [그림 6(b)]의 4)와 5)는 이를 반영한다.

Ⅶ. 평가 및 비교 분석

7.1. 기존 연구와의 차이점 정리

[표 3]은 본 연구의 내용과 가장 유사한 연구인 Jain과 Farkas의 연구 내용과의 차이점을 정리한 표이다. 본 절에서는 이러한 차이점이 어떤 의미를 가지는 지를 설명한다. 먼저 ①번의 비교에서는 제안 모델이 XML 접근 제어 [3, 4, 5] 에서와 같이 하위 노드로 전파되는 접근 권한 명세를 수행함을 보여준다. 이는 보안 관리자의 손쉬운 권한 명세를 위하여 지원되어야 할 특성이다. 따라서 ②번에서와 같이 권한 충돌의 문제가 접근 권한 명세시의 하위 노드로의 권한 전파와 추론시의 상위 노드로의 권한 전파 사이에서의 권한 충돌 문제로 새롭게 해석되어진다. 본 연구에서는 이를 *sign* ∈ {+, -} 값에 기반하여 [그림 3]의 4가지의 권한 충돌 유형으로 분석하였다. 특별히 본 연구에서는 ③번의 비교에서처럼 그래프 레이블링 기법을 이용하여 조상/후손 관계의 권한 충돌을 효율적으로 발견하는 방법을 제시하였다. 제시된 방법은 Jain과 Farkas의 방법을 기본으로 하여 조상/후손 관계의 권한 충돌 관계를 더욱 효율적으로 발견할 수 있도록 하는 것이다. RDFS entailment [13]의 다양한 추론중 핵심되는 추론은 subClassOf, subPropertyOf와 같은 포함 관계 (subsumption) 기반 추론이므로, 추

〔표 3〕 본 연구의 내용과 기존 연구와의 차이점 분석

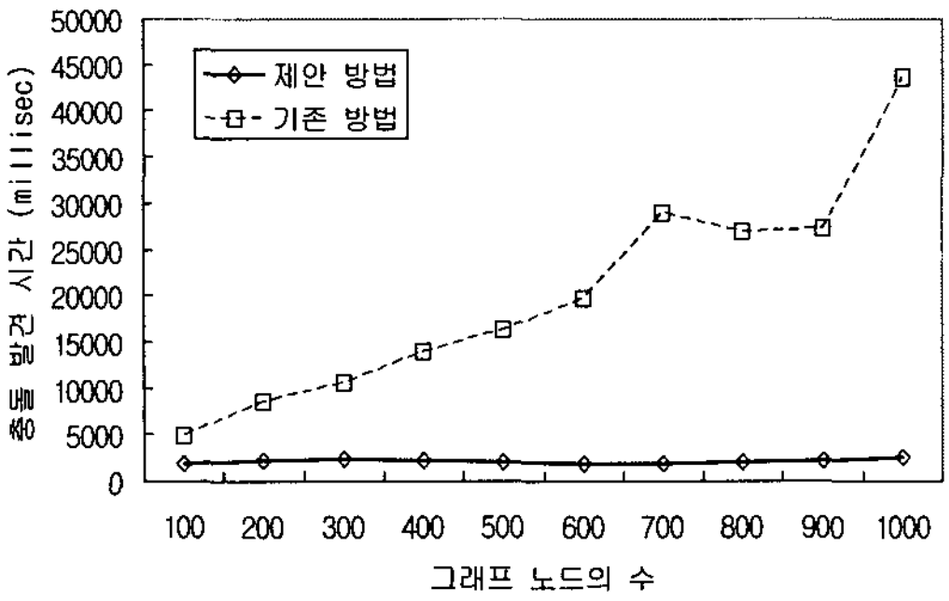
비교 기준	본 연구의 RDF 접근 제어	Jain & Farkas [6]
① 권한 명세 모델의 정의	- 권한 명세시의 하위 클래스, 하위 속성으로의 권한 전파 정의 - 정의 4의 a = <*subject, object, action, sign, type*>에서 *type* ∈ {L, R}에 따른 권한 전파 정의 - 정의 5의 RDF * 패턴에 따라 클래스 단위의 하위 클래스로의 권한 전파 소개 - IS_A 관계를 갖는 인스턴스로의 권한 전파 정의	- 없음.
② 추론에 의한 권한 충돌 문제	- 포함 관계 기반 추론 (subClassOf와 subPropertyOf 추론) 및 범위 추론에서의 권한 충돌 문제에 대한 구체적 소개 - 권한 충돌 문제를 권한 명세에서의 권한 전파와 추론에 의한 권한 전파 두 개념 사이에서의 충돌 문제로 분석 - 권한 전파의 네 가지 유형 분류에 따른 충돌 분석	- RDF Entailment [13]에 제시된 모든 추론을 고려 - RDF 트리플에 기반하여 RDF 추론에 의한 권한 충돌의 문제를 처음으로 소개
③ 권한 충돌 발견 방법	- 네 번째 충돌 유형 "*ci* ⊂ *cj, caj-* → *cai-, cai+'* => *caj+*'"에 따른 조상 클래스 혹은 속성이 (-)의 접근 권한을 가지고, 후손 클래스 혹은 속성이 (+)의 접근 권한을 가지는 경우만을 조사 - 그래프 레이블링 기법을 이용한 조상/후손 관계 권한 충돌의 효율적 파악	- 모든 인스턴스에 대하여 조사하는 단순한 충돌 발견 알고리즘 제시
④ RDF 웹 문서의 정보 은닉에 응용	- 분석된 RDF 권한 충돌 문제를 웹 문서의 정보 은닉에 연관하여 소개	- 없음.

론에 의한 권한 충돌 발견을 전반적으로 향상시킬 수 있다.

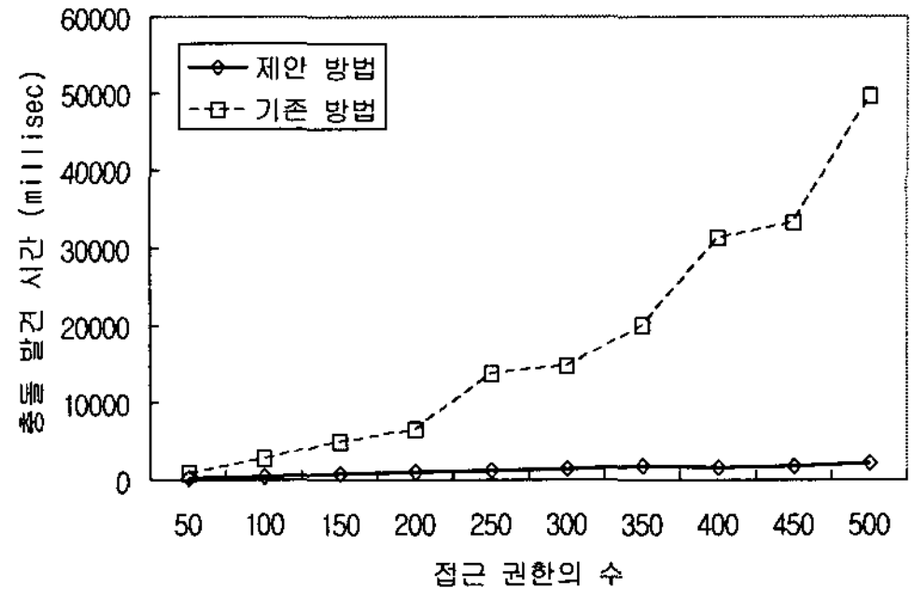
7.2. 제안된 그래프 레이블링을 통한 권한 충돌 발견 방법의 효율성 분석

본 절에서는 Jain과 Farkas가 제안한 모든 인스턴스에 대한 단순한 권한 충돌 조사 방법과 본 연구에서 제안한 그래프 레이블링을 이용한 해당 조상 또는 후손 노드에 대한 접근 권한만을 조사하는 방법에 대한 몇 가지 성능 평가 결과를 보인다. 실험 데이터 및 접근 권한의 생성을 위하여, 임의의 DAG (Directed Acyclic Graph)를 그래프 노드의 수 (즉, 클래스 혹은 인스턴스의 수)와 한 클래스당 속성의 수에 따라 랜덤하게 생성하였다. 그래프 노드의 수는 100에서 1,000개까지 실험하였으며, 한 노드당 속성의 수는 1에서 5까지 랜덤하게 발생하도록 프로그래밍 하였다. 생성된 DAG에 대하여 임의의 접근 권한들을 순차적으로 생성하여, 이전에 생성된 접근 권한과의 충돌 관계를 갖는 지를 실험하였다. 접근 권한의 수는 500개까지 실험되었으며, 충돌 유형은 subClassOf 관계에 대해서만 수행되었다. 이는 제안하는 방법이 조상/후손 관계의 권한 충돌 조사의 이점을 활용하는 것이므로, subClassOf 관계에 대한 실험이 subPropertyOf 관계에 대한 실험 결과를 또한 반영하기 때문이다. 모든 실험은 RAM 1GB 및 3.2 GHz 듀얼 팬티엄 Ⅳ CPU를 가진 윈도우 XP 컴퓨터에서 수행되었으며, 그래프 레이블링 알고리즘으로 소수 기법 [11]을 사용하였다.

먼저 [그림 8 (a)]의 그래프는 RDF/RDFS에서의 그래프 노드의 수, 즉, 클래스, 속성, 혹은 인스턴스의 수에 따른 권한 충돌 발견 시간을 비교한다. 그래프의 노드 수가 많은 것은 정의된 클래스 혹은 속성의 수, 또한 인스턴스의 수가 많다는 것을 의미한다. Jain과 Farkas의 방법은 모든 RDF 인스턴스를 조사하는 것이므로, 노드 수가 증가할수록 발견 시간이 매우 커지는 것을 관찰할 수 있다. 또한 본 연구에서 제안한 발견 방법은 subClassOf 추론에 있어 (-)/(+)의 조상/후손 관계의 접근 권한들만을 조사하는 것이므로 노드 수가 증가하더



(a) 클래스, 속성, 혹은 인스턴스의 수에 따른 충돌 발견 시간



(b) 접근 권한의 수에 따른 충돌 발견 시간

〔그림 8〕 Jain 과 Farkas의 기존 방법과의 권한 충돌 발견 시간 비교

라도 그 증가율이 현저히 낮음을 관찰할 수 있다. 즉, RDF 인스턴스의 수에 큰 영향을 받지 않음을 알 수 있다. 다음으로 [그림 8(b)]의 그래프는 명세되는 접근 권한의 수에 따른 권한 충돌 발견 시간을 비교한다. 마찬가지로, 접근 권한의 수가 증가할수록 Jain과 Farkas의 방법에 비교하여 제안된 방법은 매우 낮은 발견 시간 및 증가율을 가짐을 확인할 수 있다.

Ⅷ. 결 론

최근 웹을 보다 더 지능적으로 만들기 위한 노력과 함께 시맨틱 웹에 대한 많은 연구가 진행 중이다. 하지만 그와 관련한 보안 문제에 관한 연구는 미비한 실정이라고 판단된다. 본 연구에서는 이러한 목적으로, RDF 보안과 관련한 RDF 접근 제어 명세 모델 및 RDF 추론에서의 접근 권한 충돌 문제를 자세히 분석 소개하였다. 비록 Jain과 Farkas 또한 RDF 트리플에 기반한 추론 시의 권한 충돌 문제를 소개하였지만, 본 연구에서는 그들과 다르게 접근 권한 충돌 문제를 접근 권한 명세와 추론 사이에서의 권한 충돌 문제로 새롭게 분석 소개하였다. 즉, 접근 권한 명세시의 subClassOf 관계 및 subPropertyOf 관계에 있어 하위 클래스, 하위 속성으로의 권한 전파를 정의하였고, 또한 반대로 RDF 추론시의 subClassOf, subPropertyOf 관계에 있어 상위 클래스, 상위 속성으로의 권한 전파를 정의하였다. 이러한 정의에 기반하여 두 권한 전파 시의 권한 충돌 유형을 분석하였으며, 이를 토대로 그래프 레이블링 기법을 이용한 효율적 권한 충돌 발견 방법을 제시하였다. 특별히 제안된 접근 제어 모델은 RDF 웹 문서의 세밀한 정보 은닉 문제와 연관되어 개발되었으며, 권한 충돌 문제 또한 이와 연관하여 분석되었다.

앞으로의 연구 방향은 제안하는 RDF 접근 제어 모델을 보다 복잡한 개념 관계 및 추론을 지원하는 OWL로 확장하는 것이며, 보다 복잡한 충돌 유형의 분석과 이의 효율적 발견 메커니즘을 연구하는 것이다.

참고문헌

[1] RDF Primer, W3C Recommendation, http://www.w3.org/TR/rdf-primer/

[2] OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/

[3] E. Damiani, S. D. C. Vimercati, S. Paraboschi, P. Samarati, "A fine-grained access control system for XML documents", *ACM Transactions on Information and System Security*, 5(2), pp. 169-202, 2002.

[4] E. Bertino, S. Castano, E. Ferrari, M. Mesiti, "Specifying and enforcing access control policies for XML document sources", *World Wide Web Journal*, 3(3), pp. 139-151, 2000.

[5] E. Bertino, E. Ferrari, "Secure and selective dissemination of XML documents", *ACM Transactions on Information and System Security*, 5(3), pp. 290-331, 2002.

[6] A. Jain, C. Farkas, "Secure resource description framework: an access control model", *Proc. of*

11th ACM Symposium on Access Control Models and Technologies, pp. 121-129, June 2006.

[7] L. Qin, V. Atluri, "Concept-level Access Control for the Semantic Web", *Proc. of ACM Workshop on XML Security 2003*, pp. 94-103, Oct. 2003.

[8] S. Javanmardi, M. Amini, R. Jalili, "An Access Control Model for Protecting Semantic Web Resources", *Proc. of the 2nd International Semantic Web Policy Workshop* (SWPW'06), Nov. 2006.

[9] S. Kaushik, D. Wijesekera, P. Ammann, "Policy-based dissemination of partial web-ontologies", *Proc. of the 2005 workshop on Secure web services*, pp. 43-52, Nov. 2005.

[10] P. Reddivari, T. Finin, A. Joshi, "Policy-Based Access Control for an RDF Store", *Proc. of the Policy Management for the Web Workshop*, pp. 78-83, May. 2005.

[11] G. Wu, K. Zhang, C. Liu, J. Li, "Adapting Prime Number Labeling Scheme for Directed Acyclic Graphs", *DASFAA 2006*, pp. 787-796, April 2006.

[12] V. Christophides, G. Karvounarakis, D. Plexousakis, M. Scholl, S. Tourtounis, "Optimizing taxonomic semantic web queries using labeling schemes", *Journal of Web Semantics*, 11(1), pp. 207-228, Nov. 2003.

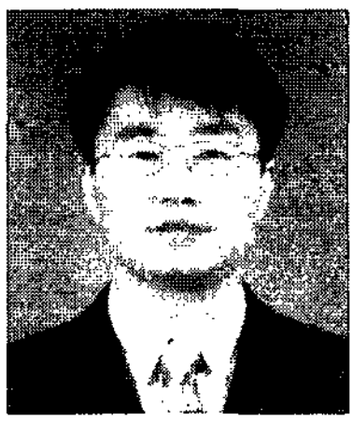
[13] RDF Semantics, W3C Recommendation, http://www.w3.org/TR/rdf-mt/

부록 1. Weapon RDF Schema 웹 문서

```
〈?xml version="1.0"?〉
〈rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:ex="http://example.org/schemas/weapons#"〉
〈rdf:Description rdf:ID = "Weapon"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/01/ rdf-schema#Class"/〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "manufacturedBy"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Weapon"〉
        〈rdfs:range rdf:resource =
"ex:WeaponCompany"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "WCode"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Quantity"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "SpecialWeapon"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource = "ex:Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "SWQuantity"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource =
"ex:SpecialWeapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "locatedIn"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/ 02/22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource =
"ex:SpecialWeapon"〉
        〈rdfs:range rdf:resource = "ex:Location"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Location"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Country"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Location"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "State"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Location"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "NuclearWeapon"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource =
"ex:SpecialWeapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "NWQuantity"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource =
"ex:NuclearWeapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "ConventionalWeapon"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource = "ex:Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "CWQuantity"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource =
"ex:ConventionalWeapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "ownedBy"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource =
"ex:ConventionalWeapon"〉
```

```
        〈rdfs:range rdf:resource = "ex:Person"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Person"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Gun"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource =
"ex:Conventional Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Rifle"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000/ 01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource =
"ex:Conventional Weapon"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "RQuantity"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/1999/02/ 22-rdf-syntax-ns#Property"/〉
        〈rdfs:domain rdf:resource = "ex:Rifle"〉
〈/rdf:Description〉
〈rdf:Description rdf:ID = "Missile"〉
        〈rdf:type rdf:resource =
"http://www.w3.org/2000 /01/rdf-schema#Class"/〉
        〈rdfs:subClassOf rdf:resource =
"ex:Conventional Weapon"〉
        〈rdfs:subClassOf rdf:resource =
"ex:SpecialWeapon"〉
〈/rdf:Description〉
〈/rdf:RDF〉
```

〈著者紹介〉



김 재 훈 (Jaehoon Kim) 정회원

1997년 2월 : 건국대학교 전자계산학과 졸업
1999년 2월 : 건국대학교 컴퓨터.정보통신공학과 석사
2005년 2월 : 서강대학교 컴퓨터학과 박사
2005년 3월~2006년 9월 : 삼성전자 정보통신총괄 통신연구소 책임연구원
2006년 9월~현재 : 서강대학교 컴퓨터공학과 BK 21 계약교수
<관심분야> 시맨틱 웹, 웹 데이터베이스, 데이터마이닝, 데이터베이스 보안

박 석 (Seog Park) 종신회원

1978년 2월 : 서울대학교 계산통계학과 졸업
1980년 2월 : 한국과학기술원 전산학과 석사
1983년 8월 : 한국과학기술원 전산학과 박사
1983년 9월~현재 : 서강대학교 컴퓨터학과 교수
1997년 2월~현재 : 한국정보보호학회 이사
1998년 9월~현재 : 데이터베이스 연구회 운영자문위원
2006년 : 국세청 정보화 자문위원
2005년 : 한국정보과학회 부회장
2004년 1월~2005년 12월 : 한국정보과학회지 편집위원장
2002년~2004년 : University of Virginia 방문 교수
2006년 : VLDB Panel Co-chair
2006년 : KCC 2006 한국컴퓨터종합학술대회 프로그램위원장
1999년~2007년 8월 : DASFAA Steering Committee
2004년 : DASFAA 2004 Organization Chair
<관심분야> 데이터베이스 보안, 실시간 시스템, 트랜잭션 관리, 데이터웨어하우스, 웹 데이터베이스