韓國 컴퓨터情報學會 論文誌
第12卷 第1號, 2007. 3.

2007-12-1-1-2

시멘틱웹 구축을 위한 스키마 관리 기법 연구

김 병 곤*, 오 성 균**

Schema management skills for semantic web construction

Byung-Gon Kim*, Sung-Kyun Oh**

요 약

전 세계적으로 인터넷의 사용이 일반화 되면서 인터넷상의 정보의 양이 기하급수적으로 증가하였고, 이에 따라 이러한 정보들을 수집하고 통합하여 특정집단 또는 일반인들의 의사결정을 지원하기 위한 시멘틱웹에 대한 중요성이 갈수록 증대되고 있다. 시멘틱웹을 구성하는 기본 구조는 온톨로지이며, XML, RDF/RDF스키마, OWL 같은 언어들은 온톨로지의 스키마를 구성하는 기본 수단이다. 온톨로지의 스키마를 구성하고 관리할 때 중요한 고려사항 중 하나는 스키마는 시간이 지남에 따라 변화한다는 것이다. 그러므로 스키마상의 도메인의 변화, 데이터 개념의 변화 혹은 자원간의 관계의 변화 등을 감지하고 이를 반영할 수 있는 형태로 구현되어야 한다. 본 연구에서는 시멘틱웹의 스키마관리를 위한 버전 관리 기법을 제안한다. 이를 위하여 버전의 변화 형태를 카테고리 별로 분류하고 이를 바탕으로 버전 그래프를 생성하였다. 생성된 그래프를 바탕으로 이행성규칙 등을 정의하여 적용하였으며, 좀 더 세세한 적용이 가능하도록 표식을 사용하여 적용 가능한 버전 스키마의 범위를 확장하도록 하였다.

Abstract

As the information of the Internet increased, importance of sematic web for collecting and integration of these informations to support decision making of some group or ordinary people are growing as well. Basis structure that composes semantic web is ontology and languages like XML, RDF/RDF schema and OWL are basis means that compose ontology schema. When composes and manages Ontology schema, one of the important consideration point is that schema is changed as times go by. Therefore, change of domain of schema, change of data concept or change of relation between resource etc. are reflected in the ontology system. In this study, we suggest semantic web schema management skill in terms of version management. We categorized version change forms and created version graph for checking of version transition. With created version graph, we define transitivity rule and propose schema tag for detail application which enables extending of applicable version schema.

▸ Keyword : 시멘틱웹(semantic web), 온톨로지(ontology), 온톨로지 버전(ontology version), 이행성규칙(trasitivity rule), RDF 스키마(RDF schema)

• 제1저자 : 김병곤
• 접수일 : 2007.1.9, 심사일 : 2007.1.22, 심사완료일 : 2007. 3.12.
* 부천대학 e-비즈니스과 조교수 **서일대학 소프트웨어과 교수
※ 본 논문은 서일대학 과학기술연구소의 연구지원금에 의한 연구실적물임.

Ⅰ. 서 론

인터넷은 항상 변한다. 즉, 실제 인터넷 환경에서 같은 자원에 대한 다른 버전의 정보들이 존재한다. 그러므로 인터넷 시스템은 이러한 변화를 다룰 수 있는 관리 기법을 필요로 한다. 시멘틱웹에서 온톨로지를 구성하는 메타데이터 스키마의 버전관리는 메타데이터 스키마의 새로운 변화를 감지하고 변화에 대하여 저장 관리하여 좀 더 정확한 정보를 얻고자하는 사용자의 요구에 대응하는 것이다. 이를 위하여 먼저, 스키마의 변화에 대한 인식과 정의가 필요하며 이를 바탕으로 버전을 정의하고 구분할 수 있어야 한다. 온톨로지 연구 초기에는 XML을 바탕으로 연구가 진행되었으며, XML에 대한 버전 연구가 부분적으로 진행되었다[1]. 온톨로지를 표현하기에 적합한 RDF/S(RDF 스키마)와 같은 메타데이터 스키마 언어에 대한 버전 연구는 단순한 XML과는 달리 여러 가지 측면에서 연구되고 있다. [2]에서는 온톨로지의 버전에 관한 전반적인 개념을 정의하고 연구방향을 소개하였다. [3]에서는 복수의 온톨로지를 합치는 머징에 대한 연구를 수행하였고, Ontoview와 Prompt 시스템의 연구에서는 버전간의 차이를 감지하고 분석하기 위한 여러 가지 함수에 대한 연구를 수행하였다.[4, 5] Semversion 시스템은 RDF를 기반으로 한 온톨로지 버전 시스템을 구현하는 것을 목표로 머지, 충돌 감지 등의 방법론을 제시하였다[6].

시멘틱웹 상의 데이터는 생성될 때 현재의 스키마의 버전에 맞추어 정의되고 생성된다. 그러나 버전의 변화에 따라 현재의 스키마 버전과 상이한 부분이 발생하면 질의를 처리할 때 문제가 발생할 수 있다. 이때, 데이터와 버전간의 정확한 관계를 명시해주어야 최신의 정확한 시멘틱웹을 구축 할 수 있다. 본 연구는 이처럼 스키마의 변화에 따른 데이터의 적용 스키마를 정확히 파악하고 결정하기 위한 버전 선택 스키마 관리기법을 제시하였다. 특히 다른 연구에서 다루지 않은 적용 가능한 버전의 확대를 위한 방안을 중심으로 기술하였다. 메타데이터 스키마로는 온톨로지를 구성하는데 가장 보편적으로 사용되고 있는 RDF/S를 대상으로 하였다. 먼저, 버전의 변화 형태를 카테고리 별로 분류하고 이를 바탕으로 버전 그래프를 생성하였다. 생성된 그래프를 바탕으로 이행성규칙 등을 정의하여 적용하였으며, 좀 더 세세한 적용이 가능하도록 표식을 사용하여 적용 가능한 버전 스키마의 범위를 확장하도록 하였다.

Ⅱ. 관련연구

2.1 메타데이터 스키마 언어

시멘틱 웹은 웹 상에 정보의 리소스들이 서로 의미적 연결을 가지고 있고 인간과 기계 모두가 쉽게 문서를 이해할 수 있도록 지원해야 한다. 초기의 연구들에서는 XML을 기반으로 한 연구들이 많이 진행 되었으나, XML의 구조적인 한계와 표현능력의 한계로 인하여 점차 RDF[7], RDF/S[8], OWL[9]등의 시멘틱웹 전용언어로의 연구로 변화하고 있다.

온토롤지를 표현하기 위해 W3C에서 제안된 RDF는 웹 상에 존재하는 자원들의 메타데이터를 기술하기 위한 언어이다. RDF는 XML 문법을 따르고 있으며, 기본적으로 자원-속성-값의 기본 구조를 이용하여 자원의 메타데이터를 기술한다. RDF/S는 RDF를 기술하기 위해 필요한 용어들의 의미와 용어들 간의 의미적 관계를 기술하기 위해 W3C에서 제안한 언어이다. RDF/S를 이용해 정의된 용어와 용어들 간의 관계는 RDF로 기술된 메타데이터의 의미를 좀 더 명확하게 표현할 수 있다.

OWL은 기존 RDF/S를 이용한 온톨로지의 표현을 확장 및 강화하기 위하여 개발되고 있다. 또한, 표현이 대상이 되는 클래스, 속성, 인스턴스 간의 복잡한 관계에 대한 기술이 가능하기 때문에 RDF/S를 이용한 질의 유형 보다 확장된 형태의 다양한 질의 처리가 가능하다. 즉, 온톨로지의 다양한 표현에 대한 보다 많은 제약사항을 기술하기 때문에 클래스와 구성원간의 관계를 집합이란 개념을 통하여 온톨로지로는 정의되지 않는 사실들에 대한 논리적인 추론이 가능하기 때문에 최근 이를 이용한 다양한 연구가 진행되고 있다.

위에서 언급된 메타데이터 스키마 언어들은 현재 활발히 진행되고 있는 OWL-S[10]나 WSMO[11]와 같은 시멘틱 웹 서비스를 위한 온톨로지 표준 모델을 구성하는 중요한 요소로 사용되고 있다. 본 논문에서는 시멘틱웹 언어 중 온톨로지 메타데이터 스키마를 기술하기 위해 초창기에 제안되어 현재 많은 연구들이 진행되고 있고, 응용 프로그램들의 지원이 가능한 RDF/S를 대상으로 스키마의 변화에 따른 스키마 관리 연구를 진행하였다.

2.2 메타데이터 스키마의 변화와 호환성

RDF 스키마의 변화는 데이터 도메인의 변경, 데이터 개

념의 변화, 데이터 자원간의 관계의 변화 등을 내포하고 있다. 더욱 자세히 표현하면 스키마에 새로운 요소나 애트리뷰트 등이 추가될 수도 있으며, 기존의 요소나 애트리뷰트가 제거되는 변화가 발생할 수 도 있다. 또한, 특정 클래스의 영역(Range)이나 도메인(Domain)이 변경 될 수도 있다. SubClassOf, subPropertyOf 와 같은 클래스나 프로퍼티간의 관계가 변화할 수 도 있다.

RDF 스키마의 변화를 다음과 같이 Add, Delete, Update, MC(Minor change) 와 같은 네 개의 카테고리로 분류할 수 있다. 먼저 Add 카테고리는 스키마에 클래스나 속성이 새롭게 추가되거나, subClassOf , SubPropertyOf 와 같은 클래스간 혹은 속성간의 관계가 추가되는 것을 의미한다. Delete 카테고리는 스키마로부터 클래스를 제거하거나, 클래스로부터 애트리뷰트를 제거하는 것과 클래스나 프로퍼티간의 관계를 삭제하는 등의 변화를 의미한다. Update 카테고리는 클래스간의 계층구조의 변화, 관계의 이름 변경, 애트리뷰트의 이름 변경 등이 이에 해당한다. 마지막으로 MC는 기존의 RDF 스키마 정보에는 아무런 영향을 주지 않으면서 그 표현하는 방법만이 바뀌는 경우이다. 이러한 변화는 실제로 저장된 정보에는 변화를 주지 않기 때문에 단순히 RDF 스키마 문서에서 문맥만이 바뀌게 되는 것이다.

인터넷 상의 시멘틱웹에서 스키마의 버전을 중요하게 고려하고 다루는 이유는 웹상에 존재하는 데이터를 적용할 가장 적합한 스키마 버전을 선택하고 적용하기 위해서이다. 즉, 동일한 데이터에 대하여 우리는 이전 버전의 스키마를 적용하여 해석하고 저장할 수도 있고, 최근의 스키마를 적용할 수 도 있다.

이러한 데이터와 스키마의 관계를 두 가지 측면으로 구분할 수 있다. 만약, 이전 버전의 스키마에 적용되도록 구성된 데이터를 새로운 버전의 스키마에 적용하여 사용하게 된다면, 이러한 경우를 스키마의 전진적 사용(Prospective use)이라고 정의할 수 있다. 반대로 새로운 버전의 스키마에 적용되도록 구성된 데이터를 이전 버전의 스키마에 적용하여 사용하게 된다면, 이러한 경우를 스키마의 후진적 사용(Retrospective use)이라고 정의 할 수 있다.[2]

이제, 전진적 사용과 후진적 사용의 두 가지 관계를 메타데이터 스키마의 변화 4가지 카테고리 측면에서 살펴보면 두 가지의 사용 방법이 항상 가능한 것은 아니다. 예를 들어, 스키마가 Delete 카테고리에 해당하는 변화가 있는 경우에는 데이터를 새로운 스키마에 전진적으로 적용할 수 없다. 즉, 웹상의 데이터가 삭제된 부분의 스키마에 적용되는 부분을 지니는 경우에 데이터와 스키마간의 불일치가 발생하기 때문이다. 이처럼 스키마 버전의 변화 카테고리에 따른 데이터에 대한 적절한 버전의 적용일치성을 호환성이라 한다.

표 1 스키마 버전의 변화에 따른 호환성 규칙
Table 1 Compatibility Rule of schema change

카테고리	호환성 규칙	
	전진적 사용	후진적 사용
Add	가능	불가능
Delete	불가능	가능
Update	불가능	불가능
Minor change	가능	가능

표 1에서 보는 바와 같이 스키마가 Add 카테고리에 해당하는 변화가 있는 경우에는 데이터에 대한 스키마의 전진적사용은 가능하지만 후진적 사용은 불가능하다. 반대로 스키마가 Delete 카테고리에 해당하는 변화가 있는 경우에는 데이터에 대한 스키마의 전진적 사용은 불가능하지만 후진적 사용은 가능하다. Update 카테고리의 경우에는 어떠한 경우에도 적용이 불가능하고, MC의 경우에는 모두 다 가능하다. 이러한 호환성 규칙은 데이터에 대한 버전의 선택에 중요한 척도가 된다.

III. 이행성규칙과 표식을 이용한 스키마버전 관리기법

3.1 버전 이행성 규칙

스키마의 버전은 한번만 변화하는 것이 아니고, 여러 번 변화할 수 있으며, 다양한 형태로 변화 할 수 있다. 예를 들어, 어떤 스키마가 두개의 다른 버전으로 각각 변화 할 수 있다. 하나는 Add 카테고리에 해당하는 변화를 하고, 다른 하나는 Delete 카테고리에 해당하는 변화를 할 수 있다. 또한 여러 단계에 걸쳐서 변화를 일으킬 수 도 있다. 이 장에서는 버전이 여러 단계에 걸쳐 변화할 때 변화 카테고리에 따른 이행성 규칙을 제시한다. 여기서 제시하는 이행성 규칙은 앞 장에서 언급한 전진적 사용과 후진적 사용의 확대 적용을 위하여 필수적인 요소가 된다.

이행성 규칙의 표현을 위하여 스키마 변화 카테고리를 다음과 같이 표시하고자 한다.

-mc-〉 : MC에 의한 버전 변화
-add-〉 : Add 카테고리에 해당하는 버전 변화
-del-〉 : Delete 카테고리에 해당하는 버전 변화
-up-〉 : Update 카테고리에 해당하는 버전 변화

예를 들어 V1 -add-〉V2 와 같이 표시되면 스키마 버전 V1이 add 카테고리에 해당하는 변화를 가지고 스키마 버전 V2로 변형되었다는 의미이다. 다음은 이행성 규칙이다.

Rule 1) If V1 -mc->V2 and V2 -mc->V3 then V1-mc->V3
Rule 2) If V1 -mc->V2 and V2 -add->V3 then V1-add->V3
Rule 3) If V1 -mc->V2 and V2 -del->V3 then V1-del->V3
Rule 4) If V1 -mc->V2 and V2 -up->V3 then V1-up->V3
Rule 5) If V1-add->V2 and V2 -mc->V3 then V1-add->V3
Rule 6) If V1-add->V2 and V2 -add->V3 then V1-add->V3
Rule 7) If V1 -add->V2 and V2 -del->V3 then V1-up->V3
Rule 8) If V1 -add->V2 and V2 -up->V3 then V1-up->V3
Rule 9)If V1 -del->V2 and V2 -mc->V3 then V1-del->V3
Rule 10) If V1 -del->V2 and V2 -add->V3 then V1-up->V3
Rule 11) If V1 -del->V2 and V2 -del->V3 then V1-del->V3
Rule 12) If V1 -del->V2 and V2 -up->V3 then V1-up->V3
Rule 13) If V1 -up->V2 and V2 -mc->V3 then V1-up->V3
Rule 14) If V1 -up->V2 and V2 -add->V3 then V1-up->V3
Rule 15) If V1 -up->V2 and V2 -del->V3 then V1-up->V3
Rule 16) If V1 -up->V2 and V2 -up->V3 then V1-up->V3

위에 제시된 이행성 규칙을 이용하면 다양한 형태의 스키마 버전 변화에 대하여 호환성 규칙을 적용 가능하게 되어, 전진적 사용과 후진적 사용의 가능 여부를 판단 할 수 있다. 예를 들어, 규칙 6번에서와 같이 버전 V1이 Add 카테고리의 변화로 V2가 되고 다시 V2가 Add 카테고리의 변화로 V3가 되었다면, V1에 적용되었던 데이터는 스키마 버전 V2와 V3에 대하여 모두 전진적 사용이 가능하다. 앞장에서 설명한 호환성 규칙과 함께 이행성 규칙을 적용하면 데이터에 대하여 적용가능한 스키마 버전을 알수 있다.

3.2 RDF 스키마 버전 그래프

온톨로지 시스템에서 스키마의 변화에 따라 데이터의 적용 스키마가 변화할 수 있다. 즉, 스키마의 호환성 법칙을 위배하지 않는 범위에서 이전 버전의 스키마를 적용할 수도 있고, 새로운 버전의 스키마를 적용할 수 도 있다.

이와 같은 버전의 선택을 위하여 버전의 변화를 기록하고 추적할 수 있도록 스키마 버전 그래프를 제안한다. 이 스키마 버전 그래프는 버전의 변화, 변화의 카테고리 등을 기록한다. 스키마 버전 그래프를 다음과 같이 정의한다.

$$G = (V, E)$$

V 는 버전을 나타내는 노드의 집합이며, E 는 스키마의 변화의 카테고리를 표시하는 에지(edge)의 집합이다.

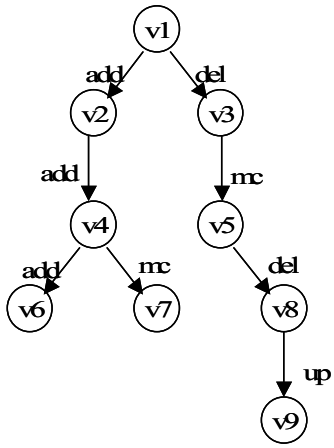


그림 1. 스키마 버전 그래프의 예
Fig 1. Example of schema version graph

그림 1은 스키마버전 그래프의 예이다. 최초의 스키마 버전 v1은 각각 버전 v2와 v3로 변화 되었다. 이때, v2는 add 카테고리에 해당하는 변화를 하였고, v3는 delete 카테고리에 해당하는 변화를 하였다. 이와 같은 방법으로 v1은 v9까지 변화한 것을 알 수 있다.

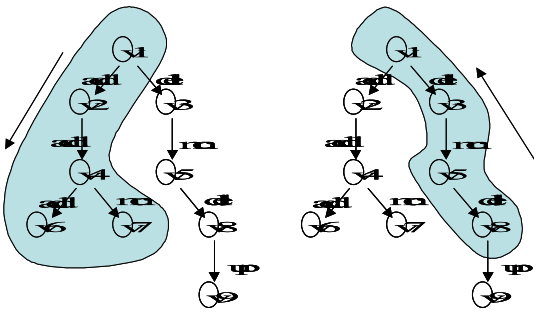
스키마 버전 그래프는 앞장에서 제시되었던 호환성 규칙과 버전간의 이행성 규칙을 적용하여 특정 데이터에 대하여 가장 적합한 버전을 선택하는데 사용된다. 예를 들어 스키마 버전 v1에 적용되도록 작성된 데이터는 v2, v4, v6에도 전진적으로 아무런 문제없이 적용 가능하다. 이는 v1이 v2, v4,v6에 이르기 까지 모두 add 카테고리에 해당하는 변화를 하였고, 이행성 규칙 6번을 적용하고, 표 1에서 add 카테고리의 전진적사용이 가능하기 때문이다. 다음은 설명을 식으로 표현한 것이다.

if v1--add->v2 and v2--add->v4 then v1--add->v4
if v1--add->v4 and v4--add->v6 then v1--add->v6

=> add 카테고리는 전진적 사용이 가능하므로 v1에 적용가능한 데이터는 스키마 v2, v4, v6에 적용가능하다.

v8에 적용되도록 작성된 데이터가 존재한다면, v1, v3, v5에 후진적으로 모두 적용가능하다. 이는 이행성 규칙 3번과 11번을 적용하고 표 1에서 delete 카테고리의 후진적 사용이 가능하기 때문이다. 다음은 설명을 식으로 표현한 것이다.

if v1—del->v3 and v3—mc->v5 then v1—del->v5
if v1—del->v5 and v5—del->v8 then v1—del->v8
=> del 카테고리는 후진적 사용이 가능하므로 v8에 적용가능한 데이터는 스키마 v1, v3, v5에 적용가능하다.



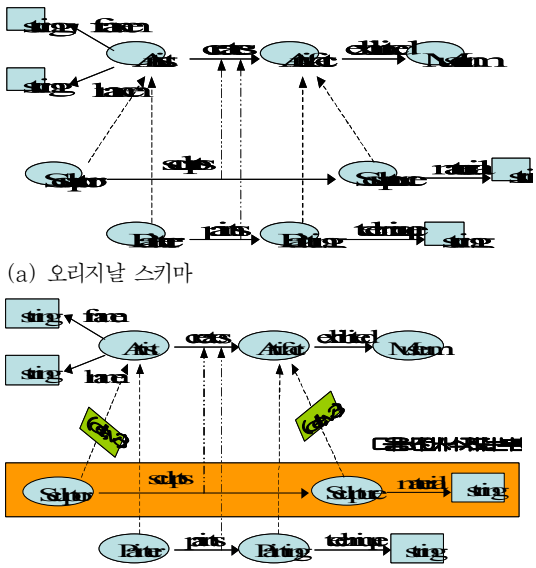
(a) v1 스키마 데이터의 전진적 적용(왼쪽)
(b) v8스키마 데이터의 후진적 적용(오른쪽)
그림 2. 이행성 규칙에 의한 적용 가능 버전의 범위
(a) Prospective use of v1 schema(Left)
(b) Retrospective use of v8 schema(Right)
Fig 2. Possible version of transitivity rule

그림 2는 앞에서 설명한 이행성 규칙 적용의 예를 그림으로 표시한 것이다. 위의 결과를 보면 하나의 데이터소스는 여러 스키마 버전에 적용이 가능한데, 이처럼 이행성 규칙과 호환성 규칙은 가능성을 제시하고 있다. 그러나, 어떤 버전의 스키마를 적용할지는 별도의 문제이며 시스템별로 별도의 원칙이나 규칙을 두는 방법이 있다.

3.3 표식을 이용한 Delete, Update 카테고리 버전 사용

표1에서 나타난 바와 같이 Delete와 Update 카테고리의 경우에는 버전의 전진적 사용이 불가능하다. 그러나, 버전의 변화에 있어서 데이터에 대한 버전의 전진적 적용이 항상 불가능한 것은 아니다. 많은 데이터들이 생성될 때 스키마의 일부분을 적용하여 생성되는 경우가 많다. 즉, 버전의 변화가 데이터의 내용과는 무관한 부분에서 발생할 수 있다는 것이다. 이러한 경우에는 버전이 어떤 형태로 변화하던지 상관없이 데이터를 전진적으로 적용 가능하다. 한번 전진적으로 사용된 데이터는 앞으로의 스키마의 버전 변화에 다시 전진적으로 사용될 수 있기 때문에 버전의 변화 단위별 정보 유지는 중요한 요소가 된다.

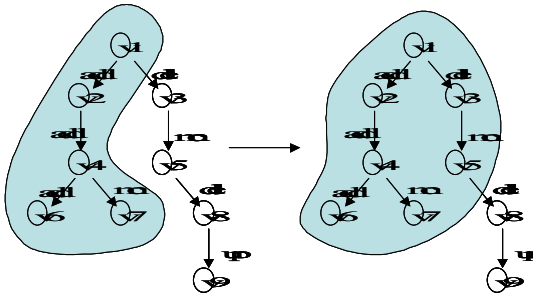
스키마에 있어서 버전에 따른 변화를 변화 단위별로 관리하기 위해서는 스키마에 표식(tag)을 사용할 수 있다. 표식은 스키마가 Delete와 Update 카테고리의 범위에서 변화한 경우에만 표시한다. 이는 Add와 Mc의 경우에는 전진적인 스키마의 적용이 가능하므로 표시하지 않는다.



(a) 오리지널 스키마

(b) 오리지널 스키마에 태그를 표시한 경우
그림 3. 태그를 이용한 버전의 변화 표시
(a) Original schema
(b) Tagged version schema
Fig 3. Tagged version representation

그림 3은 표식에 의하여 표현된 스키마의 변화를 보여주는 예이다. 그림(a)의 오리지널 스키마에서 Sculptor와 Sculpture 부분이 다음 버전에서 삭제된다면 해당하는 부분에 표식을 한다. 표식의 형식은 (del,v3)와 같은 형식으로 하게 되는데, del은 다음 버전에서 해당하는 부분이 삭제됨을 의미하고 다음 버전의 번호가 v3임을 나타낸다. 표식을 사용하게 되면 실제로 적용가능한 스키마가 확장되는 효과를 볼 수 있다. 그림4의 (a)에서 스키마 버전 v1에 적용된 데이터는 버전 그래프에 의하면 v2,v4,v6,v7에 대하여 적용 가능한 것으로 분석되었다. 그러나, 표식을 적용한 스키마와 실제 데이터를 비교하면 적용가능한 스키마가 v3, v5가 추가 될 수 있다. 이는 실제 데이터가 v3의 변화와는 무관한 경우에 가능해진다. 아래 그림은 이러한 설명을 그림으로 표현하였다.



(a) v1 스키마 데이터의 전진적 적용
(b) 표식에 의하여 적용가능한 스키마가 확장됨
그림 4. 표식적용에 의한 적용가능 스키마의 변화
(a) Prosepctive use of v1 schema
(b) Extending of applicable schema
Fig 4. Extending of applicable schema with tagged schema

IV. 결론 및 향후 과제

본 연구는 시멘틱웹의 스키마 관리를 위한 스키마 버전을 다루었다. 특히 버전이 변화함에 따라 특정 버전에 적용되었던 데이터가 최신의 다른 버전의 스키마에 적용될 수 있는지를 자동적으로 결정할 수 있는 토대를 마련하였다.

먼저 버전의 변화를 4가지의 카테고리로 분류하여 버전의 변화에 따른 가능성을 별도로 분류하였다. 이를 버전의 호환성 규칙으로 정의하였으며, 이는 버전의 전진적 사용과 후진적 사용을 모두 고려하였다. 최종적인 버전의 선택을 위하여 이행성규칙의 적용 방법과 스키마표식 방법을 사용하였다. 이행성규칙은 스키마버전그래프와 함께 적용가능한 버전의 리스트를 파악하는데 사용된다. 스키마표식은 Delete, Update 카테고리의 전진적 사용과 같은 불가능한 경우에도 가능성을 열어 최신의 스키마 버전을 사용하도록 하기 위하여 사용되는 방식이다.

본 연구에서 제시된 내용들은 RDF스키마와 같은 온톨로지 스키마를 사용하고 버전간의 차이를 유지하는 시멘틱웹 시스템에서 적용가능하다. 제시된 이론들은 궁극적으로 가장 최신의 버전의 스키마를 데이터에 적용 가능하도록 하는 것이 목표이다.

참고문헌

〔1〕 W3C, "Guide to Versioning XML Languages using XML Schema 1.1 Based on XML Schema 1.1 Working Draft of 31 Aug 2006", W3C Working Draft 28 September 2006

〔2〕 Michel Klein and Dieter Fensel. "Ontology versioning for the Semantic Web.", In: Proceedings of the First International Semantic Web Working Symposium (SWWS), USA , page 75-91, 2001

〔3〕 D. McGuiness, R. Fikes, J. Rice, and S. Wilder. "An environment for merging and testing large ontologies.", In Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning, 2000

〔4〕 M. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. "Ontology versioning and change detection on the web.", In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Spain, 2002

〔5〕 N. F. Noy and M. A. Musen. "PromptDiff: A fixed-point algorithm for comparing ontology versions.", In Eighteenth National Conference on Artificial Intelligence (AAAI-2002), 2002

〔6〕 Max V ölkel, Tudor Groza , "SemVersion: An RDF-based Ontology Versioning System", In Proceedings of IADIS International Conference on WWW/Internet, volume 1, Spain, pp. 195-202. 2006.

〔7〕 W3C, "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation 22 February 1999

〔8〕 W3C, "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation 10 February 2004

〔9〕 W3C, "OWL Web Ontology Language Overview ", W3C Recommendation 10 February 2004
〔10〕 W3C, "OWL-S: Semantic Markup for Web Services", W3C Member Submission 22 November 2004
〔11〕 Stollberg M., Cimpian E., Mocan A., and Fensel D., "A Semantic Web Mediation Architecture." In Proc. of the Canadian Semantic Web Symposium (CSWWS 2006), Quebec, Canada., 2006

저 자 소 개

김 병 곤

1990년 홍익대학교 공과대학 전자계산학과 졸업(이학사)
1992년 홍익대학교 공과대학 전자계산학과 대학원 졸업(이학석사)
2001년 홍익대학교 공과대학 전자계산학과 대학원 졸업(이학박사)
2001년 ~ 현재 부천대학 e-비즈니스과 조교수
〈관심분야〉 다차원 인덱싱, 시멘틱 웹

오 성 균

1981년 홍익대학교 이공대학 전자계산학과 졸업(이학사)
1984년 연세대학교 산업대학원 전자계산학과 졸업(공학석사)
1999년 홍익대학교 이공대학 전자계산학과 졸업(이학박사)
1987년 ~ 현재 서일대학 소프트웨어과 교수
〈관심 분야〉 능동 데이터베이스, XML 모델링, 소프트웨어 공학