시맨틱 웹에서 온톨로지 버전 관리

윤홍원* · 이중화**
*신라대학교, **동의대학교

Ontology Versions Management on the Semantic Web

Hongwon Yun* · Junghwa Lee**
*Silla University, **Dongeui University
E-mail : hwyun@silla.ac.kr, junghwa@deu.ac.kr

요 약

온톨리지는 시맨틱 웹의 핵심 요소이며 시간이 흐름에 따라서 변화하는 특징을 가지고 있다. 온톨리지에서 발생하는 변화는 도메인 변화, 개념 변화, 메타데이터 변화 등이 있다. 이 논문에서는 이들 변화와 시간차원을 기반으로 하는 변경 집합을 제안한다. 온톨리지 버전은 대량의 정보를 저장하고 관리해야 하는데 이를 위해서 변경 집합을 이용한 버전 저장 방법들 제안한다. 제안하는 방법들 사이에 성능을 비교하고 평가한다.

ABSTRACT

Ontology is an essential component of the semantic web and it continue to change and evolve. We study a set of changes based on domain changes, changes in conceptualization, metadata changes, and time dimension. Ontology versioning brings about massive amount of versions to be stored and maintained. We present several storage policies and conduct a set of experiments to compare the performance of each storage policies.

키워드
Semantic web, Ontology, Versions management, Storage policy

Ⅰ. 서 론

온톨로지는 시맨틱 웹의 핵심 요소이며 시간이 흐름에 따라서 변화하는 특성을 가지고 있다. 온톨로지의 변화는 도메인 변화, 개념 변화, 명세 변화가 있다[1-4]. 이 논문에서는 이들 온톨로지 변화에 근거한 변경 집합을 제안한다. 이 변경 집합에는 시간지원 온톨리지 질의를 지원하기 위해서 유효시간과 트랜잭션 시간을 포함한다. 이 논문에서 제안하는 변경 집합을 기반으로 하는 온톨로지 저장 기법을 제안하고 제안하는 저장 기법 사이에 성능을 비교하고 평가한다. 이 논문의 구성은 다음과 같다. 2장에서는 온톨로지의 변화에 대해서 살펴보고, 3장에서는 변경 집합을 제안한다. 4장에서는 이 변경 집합을 이용한 온톨로지 저장 기법을 제안한다. 마지막으로, 5장에서 저장방법들 사이에 성능을 평가한다.

Ⅱ. 온톨로지의 변화

Gruger[5]에 의하면, 온톨로지는 어떤 도메인에서 서로 공유하는 개념을 명세화한 것이라고 한다. 온톨로지에서 생기는 변화는 도메인의 변화, 공유하는 개념의 변화, 명세의 변화가 있다[4]. Wiederhold는 도메인을 다음과 같이 네 가지로 분류하였다: 용어, 영역, 인코딩, 컨텍스트. 온톨로지에 대한 연구는 정형화된 온톨로지를 정의하는 것으로부터 시작하였으며, 최근에는 온톨로지를 표현하는 언어, 온톨로지 버전을 관리하는 방법으로 연

구 관심이 이동하고 있다. 대량의 온톨로지는 웹 서비스 시스템에서 핵심 요소이며 이 대량의 데이터는 관리를 필요로 한다. 이 논문에서는 대량의 온톨로지 버전을 저장하는 방법에 대해서 살펴본다.

III. 변경 집합

변경 집합은 인스턴스 데이터 변경, 구조 변경, 식별자 변경, 메타데이터 변경이 있으며 여기에 시간차원을 추가한다. 변경 집합에 사용한 기호는 다음과 같다.

- D: 인스턴스 데이터 변경
- S: 구조 변경
- I: 식별자 변경
- M: 메타데이터 변경
- T: 시간차원

변경 집합 C는 다음과 같이 정의한다.

- C = (DUSUIUM) ◦ T

어떤 n번째 버전을 Vn이라고 하고, Vn에서 나온 변경 집합을 Cn+1이라고 하면, Vn에 Cn+1을 적용하여 Vn+1을 생성할 수 있다. 이것을 식으로 나타내면 Vn+1=Vn ◦ Cn+1 이 된다.

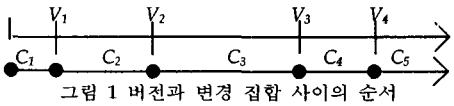

그림 1 버전과 변경 집합 사이의 순서

시간 차원 T는 유효 시간과 트랜잭션 시간을 가지며 다음과 같다.

- T = { Eid, vt, tt }
  - vt = [valid_start_time, valid_end_time]
  - tt = [transaction_start_time, transaction_end_time]

시간차원은 일반적으로 년, 월, 주, 일 등이 쓰이고 있으며 여기서는 가장 큰 시간차원을 년, 가장 작은 시간차원을 일로 한다.

- D = {Eid,ObjectName,OperationName, OldValue, NewValue}
- S = {Eid,OperationName,OldName,NewName}
- I = {Eid,OperationName,OldEid,NewEid}
- M = {Eid,ObjectName,OperatoinName,OldValue,NewValue}
- T = {Eid,vt,tt}

변경 집합에 들어가는 기본 연산은 다음과 같다. UpdateValue(e,nv,ov), DeleteValue(e), InsertValue(e,nv), RenameValue(e,ne,oe), MoveValue(e,n,p), CreateClass(e,n,p), MoveClass(e,n,p), DeleteClass(e), InsertClass(e,n,C), UnionClass(e,e1,e2), DivedeClass(e,e1,e2,n1,n2), RenameId(e,ne,oe). 앞에서 살펴 본 것처럼 어떤 버전은 변경 집합으로 생성할 수 있다.

IV. 저장 방법

앞 장에서 온톨로지의 변경 집합에 대해서 살펴보았다. 이 장에서는 온톨로지 변경 집합을 이용한 세 가지 버전 저장 방법을 살펴본다.

- 스킴 1: 변경 집합을 저장한다.
- 스킴 2: 변경 집합과 버전을 주기적으로 저장한다.
- 스킴 3: 저장 적합 시점에 변경 집합과 버전을 저장한다.

위 세 가지 저장 방법들은 기본적으로 질의 빈도가 높은 가장 최근 버전과 특정 버전을 재생성하는 데 필요한 최초 버전을 저장해 둔다.

1. 스킴 1

스킴 1은 최초 버전과 가장 최근 버전 그리고 변경 집합을 저장한다.

그림 2 스킴 1의 저장 개념

그림 2에서 V0, V1, ..., Vn은 버장을 나타내고 C1, C2, ..., Cn은 변경 집합을 의미한다. S1, S2, ..., Sn은 각 버전이 생성된 시점이고 또한 각 변경 집합이 저장되는 시점을 나타낸다.

2. 스킴 2

스킴 2는 변경 집합을 주기적으로 모아서 저장하고 변경 집합을 저장하는 주기에 버전을 저장한다. 저장 주기를 5로 가정하고 AC1, AC2, ..., AC5n/5 각각을 저장 주기마다 변경 집합의 모음이라고 하면, AC1은 첫 주기에 다섯 개의 변경 집합을 모은 것이며 AC1=C1+...+C5가 된다. 첫 주기의 저장 시점인 S1에서 AC1이 저장되고 또한 V5가 저장된다. 스킴 2에서 어떤 버전을 재생성하는 과정을 식으로 나타내면 Vq×n+1~Vq×n+(q-1) = Vq×n ◦ ACn+1(n□□0, 저장주기: q)이 된다.

3. 스킴 3

스킴 3은 저장 주기 안에서 변경 연산이 가장 많아던 버전을 저장하고, 저장하는 버전의 직전까지 변경 집합을 모아서 저장한다. 각 변경 집합 안에는 변경 집합의 연산 C(v)가 포함되어 있다. 변경 연산의 집합 C(v)를 읽어서 변경 연산이 가장 많은 버전을 찾는다 (max(count_of_UpdateOp(C1(v)),...,

Ci(v))). 변경 집합 C(v)에 들어있는 C4(v)가 가장 많은 변경 연산을 가지고 있다면 V4를 저장하고 V4까지 변경 집합의 모음 (AC1=C1,C2,C3,C4)을 저장한다. 최초 버전 V0에 첫 번째 변경 집합의 모음 AC1을 적용하여 V1,.., V3을 다시 만들 수 있으며, 이것을 V1~V3=V0 ◦ AC1로 표현한다.

V. 성능 평가

온톨로지의 크기는 100메가바이트, lifespan은 365로 하였으며, 사용자 질의의 비율은 현재 질의 50%, 과거 질의 50%로 하여 기존의 저장 방법 한 가지를 포함하여 네 가지 스킴에 대하여 실험을 하였다.

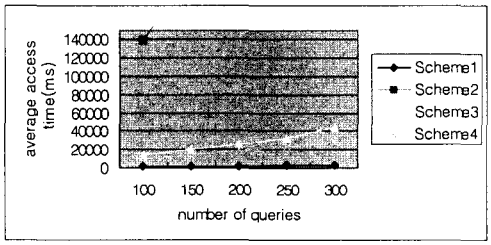

그림 3 사용자 질의에 대한 평균 응답 시간

Scheme1은 모든 버전을 저장하는 기존의 저장 방법이며 Scheme2는 본 논문에서 스킴 1에 해당하고, Scheme3은 스킴 2, Scheme4는 스킴 3에 해당한다. 스킴 1은 가장 큰 응답 시간을 보이는데, 이것은 과거 버전에 대한 질의가 들어오면 버전을 재생하는데 시간이 많이 걸리기 때문이다. 모든 버전을 저장하는 기존의 저장 방법(Scheme 1)은 버전을 재생성하는 시간이 필요 없으므로 가장 작은 응답 시간을 보이고 있으나 저장 공간 실험에서 가장 나쁜 성능을 보였다. 스킴 2와 스킴 3은 저장 공간을 작게 차지하면서 사용자 질의에 대해서 비교적 우수한 성능을 보였다.

V. 결 론

온톨로지는 시간이 흐름에 따라서 변화되며 버전을 생성한다. 대량의 버전 데이터 관리가 필요한 온톨로지의 특성을 고려하여 변경 집합을 근거로 한 온톨로지 버전의 저장 방법을 살펴 보았다. 이 논문에서는 변경 집합을 제안하였으며 이 변경 집합을 이용한 세 가지 저장 방법을 제안하였다. 기존의 방법과 제안한 저장 방법 사이에 사용자 질의에 대하여 성능을 비교하였다.

참고문헌

[1] Shu-Yao Chien, Vassilis J. Tsotras, Carlo Zaniolo, "XML Document Versioning", SIG MOD Records, Vol. 30 No. 3, pages 46-53, Sep. 2001

[2] B. Benatallah, M. Mahdavi, P. Nguyen, Q. Z. Sheng, L. Port, B. McIver, "An Adaptive Document Version Management Scheme", The 15th Conference on Advanced Information Systems Engineering (CAiSE'03), pages 16-20, Austria, 2003

[3] Benatallah, B. "A Unified Framework for Supporting Dynamic Schema Evolution in Object Databases", 18th Int. Conf. on Conceptual Modeling - ER'99, Paris, France, Springer-Verlag (LNCS series), 1999

[4] M. Klein and D. Fensel, "Ontology versioning for the Semantic Web",In Proceedings of the International Semantic Web Working Symposium (SWWS), pages 75-91, Stanford University, California, USA, 2001

[5] Sommerville, I., Rodden, T., Rayson, P., Kirby, A., Dix, A. "Supporting information evolution on the WWW", World Wide Web 1, pages 45-54, 1998