# 시맨틱 웹 데이터의 버전 관리를 위한 변경 탐지 시스템

임동혁○ 김형주
서울대학교 컴퓨터공학부
dhim@idb.snu.ac.kr, hjk@snu.ac.kr

# Change Detection System for Version Management in the Semantic Web Data

Dong-Hyuk Im○ Hyoung-Joo Kim
Dept. of Computer Science & Engineering, Seoul National University

**요 약**

시맨틱 웹이 차세대 웹으로 연구가 진행됨에 따라 그 사용이 높아지고 있다. 시맨틱 웹에서 메타 정보를 표현하는 언어로 사용되는 온톨로지는 실세계를 모델링하기 때문에 끊임없이 갱신이 이루어진다. 또한 시맨틱 웹 환경에서는 사용자가 비집중화 되어 있으며 각 사용자간의 데이터 교환이 활발하게 이루어지게 되므로 데이터의 동기화가 필요하게 된다. 따라서 데이터에 대한 버전 관리가 필요하게 되며 이를 위해서는 변경 탐지가 우선적으로 기반이 되어야 한다.

기존의 온톨로지에 대한 변경 탐지 기법은 온톨로지를 구성하는 트리플의 차집합 연산으로 이루어지는 구조적 변경을 지원하고 있으나 공노드에 대한 고려가 없으며 추론을 사용하는 변경 탐지 역시 데이터 크기와 시간의 증가라는 단점을 가지고 있다. 본 논문에서는 RDF 기반의 온톨로지 변경 탐지 시스템인 R-Diff를 제안한다. 제안된 기법은 트리플 파티션을 이용하여 공노드에 대한 레이블을 사용하며 후방향 전진 추론 기반으로 모델 일부분에만 추론을 적용하여 변경 내용을 계산한다.

## 1. 서 론

최근 차세대 웹으로 메타 데이터의 개념을 통하여 웹 문서에 의미적 정보를 추가하여 사람이 아닌 프로그램 또는 소프트웨어 에이전트가 의미 정보를 자동 추출 하는 시맨틱 웹이 주목을 받고 있다. 이러한 의미적 정보에는 특정 도메인의 개념(concept)과 그 개념 사이의 관계(taxonomy and relation)를 정의하는 온톨로지가 중요시 된다. 인공지능 분야에서 연구되어 온 온톨로지는 다양한 지식 도메인을 모델링 하기 때문에 계속해서 진화하는 특성을 가지게 된다[4]. 또한 시맨틱 웹에서의 온톨로지는 중앙 집중적이 아닌 비집중화(de-centralize)와 협업적인(collaborative) 방법으로 개발이 되므로 분산된 각 로컬 사이트에서의 동기화 및 버전 관리가 필요하게 된다[8, 11]. 온톨로지에 대한 버전 관리는 다음과 같은 기능을 제공해 주어야 한다[9].

첫째, 여러 온톨로지 버전을 저장하고 각 버전별 변경 부분을 정확하게 탐지해주어야 한다. 일반적으로 변경되는 부분은 전체 데이터의 크기에 비해 적은 양이므로 변경된 부분만 동기화 시키면 된다. 따라서 온톨로지의 변경 부분을 정확하게 찾아 주는 기법이 필요하게 된다.

둘째, 각 버전은 다른 버전으로의 변형이 가능해야 한다. 각 로컬에서 변경된 정보를 바탕으로 다른 버전으로의 변형이 가능해야 한다.

셋째, 사용자가 여러 버전에 용이하게 접근할 수 있도록 투명한 접근성을 제공해주어야 한다. 새로운 버전의 데이터와 연관 있는 이전 버전의 데이터들을 자동적으로 처리해주어야 한다. 이와 같은 기능을 위해서는 두 버전 간의 차이를 정확하게 찾아내는 효율적인 변경 탐지가 공통적으로 기반이 되어야 한다.

시맨틱 웹에서 온톨로지를 표현하는 데이터 모델로 RDF(Resource Description Framework)[6], OWL(Web Ontology Language)[7], Topic Map[3]등이 있다. 본 논문에서는 RDF 기반의 온톨로지 버전 시스템을 위한 변경 탐지 시스템 R-Diff 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 온톨로지 갱신 및 버전에 대한 관련 연구를 설명하고 3장에서는 본 논문에서 사용하는 온톨로지 모델인 RDF 모델을 설명한다. 4장에서는 제안하는 구조적 변경 탐지 기법과 함의 규칙을 이용하는 변경 탐지 기법에 대해서 자세히 설명한다. 5장에서는 결론 및 향후 연구에 대해 기술한다.

## 2. 관련 연구

80년대 중반의 객체지향 데이터베이스에서 연구되었던 스키마 진화[1]와 버전 관리는 온톨로지 진화와 버전 관리에 많은 연구 이슈를 제공해주고 있다. 특히 [1]에서 사용되는 갱신 연산에 대한 정의와 무결성 조건들을 따르는 연산들은 온톨로지 연산에도 사용되고 있으며 연구되고 있다.

또한 SemVersion[12], PromptDiff[10]와 같이 온톨로지 버전을 비교하여 변경 내용을 탐지하는 시스템들도 연구되고 있다. 본 논문의 온톨로지 변경 탐지 시스템은 이들 시스템과 유사한 기능을 가지고 있을 뿐만 아니라

RDF 모델에서 생기는 공노드와 RDF 스키마의 함의 규칙을 이용하여 향상된 변경 탐지 기법을 제공해 준다.

## 3. 온톨로지 데이터 모델

R-Diff 시스템에서는 온톨로지 모델로 RDF 모델을 사용한다. RDF는 W3C에서 제정한 것으로 웹 상에서 웹 리소스와 같은 메타 정보를 표현하는 언어로 사용된다. RDF는 주어(Subject), 술어(Predicate), 목적어(Object)의 트리플 구조를 갖는 트리플(문장)들로 구성되어 있으며 각 RDF 모델은 트리플의 집합으로 표현이 가능하다. 따라서 변경 탐지는 두 트리플 집합의 차집합으로 계산이 가능하다. 하지만 RDF나 OWL 같은 온톨로지에서는 공노드라는 특별한 형태의 노드를 사용하게 된다. 공노드는 실제하지 않거나 실제하지만 지금 당장은 URI를 부여할 정도는 아닐 때 사용되며 웹 자원의 제한 요소를 기술 할 때 문서 기술 표현상 사용할 수밖에 없다. 이런 공노드는 시스템에서 다른 자원과 구별하기 위해서 자동으로 별도의 아이디를 부여받게 되는데 이는 RDF 모델의 비교를 어렵게 만드는 요인이 된다.

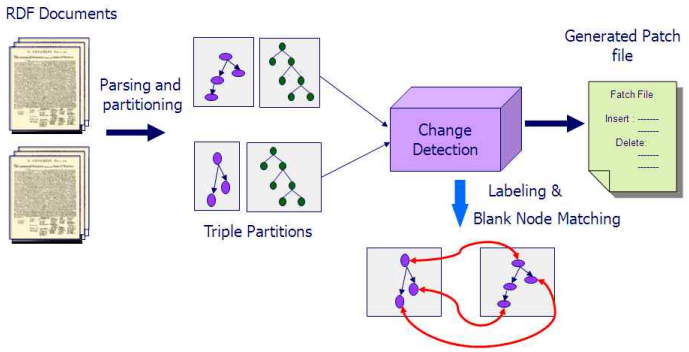


그림 1 RDF 모델의 구조적 변경 탐지

## 4. 온톨로지 변경 탐지 기법

본 논문에서 제안하는 변경 탐지 기법은 다음과 같은 2가지 기능을 제공해 준다.

### 4.1 트리플 파티션을 이용한 구조적 변경 탐지

온톨로지에 대한 구조적 변경 내용을 구한다. 기존의 LCS(Longest Common Sequence)를 사용하는 GNU Diff와 같은 툴은 단순 텍스트 비교를 통해 변경 내용을 구하게 된다. 하지만 XML, RDF 와 같은 구조적 문서에는 적용할 수 없으므로 구조적 변경 탐지가 필요하게 된다. RDF 모델은 트리플의 집합으로 그래프로 표현이 가능하므로 그래프 비교를 통한 변경 내용을 구해야 한다[2]. 구조적 변경 탐지의 개요는 그림 1과 같다. 우선 RDF 모델을 부분 트리플 집합으로 파티션한다. 이때 파티션한 부분 트리플 집합은 한 개의 자원을 루트로 가지는 트리의 형태로 표현이 된다. 따라서 같은 루트를 가지는 트리플 집합들만 비교하여 RDF 모델의 변경 탐지를 구하게 된다. 또한 공노드에 대한 매칭을 위해 2가지 공노드 레이블링 기법을 제안한다. 파티션 내에서의 노드간의 의존적 관계를 고려한 레이블과 루트에서 공노드까지의 경로를 이용한 레이블을 이용하여 공노드에 대한 매칭을 한다. 노드간의 의존적 관계는 파티션내에서 공노드가 유일하게 결정되는 경우에 해당되며 이때 공노드에 유일한 레이블을 부여할 수 있다 하지만 루트에서 공노드까지의 경로 레이블을 가지는 공노드의 경우에는 여러 개의 같은 레이블을 가진 공노드가 존재하므로 최소 크기의 공노드 매칭을 가지는 변경 탐지를 선택해야 한다. 탐지된 변경 결과는 패치 파일로 제공된다.

### 4.2 RDF 함의 규칙을 이용한 변경 탐지

온톨로지의 시맨틱을 고려한 변경 탐지 기법이다. 온톨로지에서는 개념간의 관계에 대해 추론을 사용하는데 RDF 스키마에서는 13개의 함의 규칙[5]을 이용하여 기존의 트리플로부터 다른 트리플을 추가하는 구조를 갖는다. 함의 규칙을 이용하는 변경 탐지는 단순한 구조적 변경 탐지보다 변경 내용을 줄일 수 있는 장점을 가지게 된다[13]. 예를 들면 상하위 관계의 추이적 관계를 고려하면 자원 U가 자원 V의 하위 관계이고 자원 V가 자원 X의 하위 관계이면 자동적으로 자원 U가 자원 X의 하위 관계를 갖는 것을 의미한다. RDF 모델에 추론을 한 후 구조적 변경 탐지를 계산하면 변경 내용을 줄일 수 있는 장점이 있지만 추론을 함으로써 시간적인 비용과 크기의 증가의 문제점을 발생한다. 변경 부분을 계산하기 위해서는 두 모델의 트리플 집합을 비교하여야 하기 때문에 늘어난 크기만큼 비교를 수행해야 하기 때문이다. 따라서 대용량의 RDF 온톨로지의 경우 효율성을 고려해야 한다.

본 논문에서 제안하는 방법은 미리 추론을 사용해서 변경 탐지를 하는 것이 아니라 먼저 변경 부분을 찾은 후에 변경 된 부분에 대해서만 추론을 하는 접근 방법을 취한다. 그림 2는 제안된 방법의 개요를 보여주고 있다.

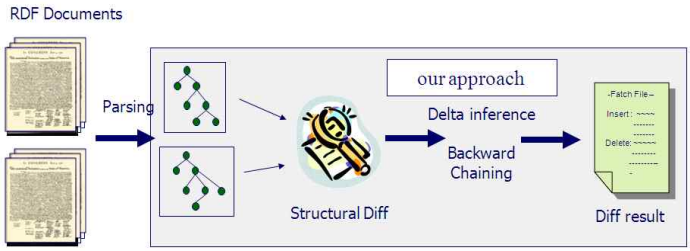


그림 2 RDF 함의 규칙을 이용하는 변경 탐지

RDF 스키마의 추론 전략은 추론 방식에 따라 전방향 추론(Forward Chaining)과 후방향 추론(Backward Chaining)으로 구분된다. 전방향 추론은 데이터를 로딩할 때 미리 추론을 계산한다. 따라서 데이터 로딩 시간이 증가하며 이에 대한 공간 비용을 많이 차지하게 된다. 전방향 추론의 장점은 미리 모든 정보를 추론하였기 때문에 빠른 질의 응답을 보인다는 것이다. 이에 반해 후방향 추론은 실행 시간에 추론을 하게 되므로 로딩 시간이 짧은 반면에 질의 응답 시간이 길게 된다. 본 논문에

서의 변경 탐지에서는 후방향 추론을 채택하며 특히 모든 트리플에 대해 추론 규칙을 적용하지 않는다. 먼저 두 모델간의 구조적인 차이를 계산된 변경 부분에 대해서만 지연(lazy) 추론을 하게 된다. 또한 변경 탐지에서 모든 추론 규칙을 적용할 필요는 없다. 상하위 관계를 나타내는 술어는 상하위 관계에 해당되는 규칙들만 적용을 하고 인스턴스에 관계된 술어에는 해당되는 규칙들만 적용을 하면 되는 것이다.

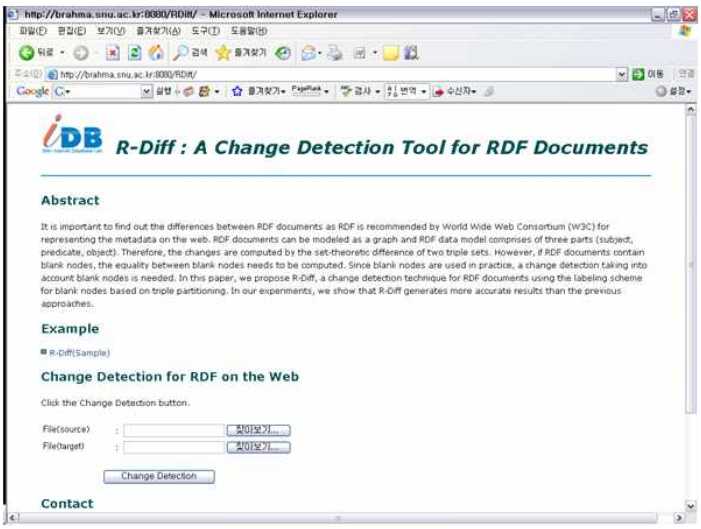


그림 3 R-Diff 시스템

그림 3은 본 논문에서 제안하는 R-Diff 시스템을 보여준다. 웹 기반의 시스템으로 2개의 RDF 문서를 입력으로 주어주면 두 문서의 변경 탐지 내용을 보여준다.

## 5. 결론 및 향후 연구

차세대 웹인 시맨틱 웹에서의 데이터는 분산된 사용자가 협업의 과정을 통해 이루어지기 때문에 각 사용자간의 동기화가 필요하며 이러한 동기화를 제공해 줄 수 있는 버전 시스템이 요구된다. 온톨로지의 버전 시스템의 기반이 되는 변경 탐지 기법은 정확하고 빠르게 탐지를 해야만 한다.

본 논문에서 제안한 R-Diff 시스템은 이러한 버전 관리 시스템에서 사용할 수 있는 변경 탐지를 제공해 준다. 시맨틱 웹과 지식 관리 시스템에서 사용되는 온톨로지에 대해 단순 텍스트 변경 탐지가 아닌 구조적 변경 탐지를 제공해 주며 또한 변경 부분에 대해서만 RDF 스키마 함의 규칙을 적용하는 효율적인 변경 탐지 기법을 제안하였다. 제안하는 기법은 구조적 변경에서 RDF 온톨로지의 특성을 이용한 트리플 파티션과 공노드에 대한 레이블링 기법을 사용하여 정확한 변경 탐지를 수행하며 후방향 추론과 트리플에 적용되는 규칙들만을 적용을 하여 기존의 전방향 추론을 이용하는 변경 탐지보다 우수한 성능을 보인다.

향후 연구로서 RDF 모델 뿐만이 아니라 OWL과 같은 다양하고 복잡한 온톨로지의 변경 탐지가 필요하다. 또한 여러 버전 사이에서의 질의 처리, 동기화 기법등에 대한 연구가 필요하다.

## 참고 문헌

[1] J. Banerjee, W. Kim, H. J. Kim, H. F. Korth, Semantics and Implementation of Schema evolution in Object-Oriented Database. In Proceedings of the SIGMOD, 1987

[2] T. Berners-Lee and D. Connolly, Delta:An Ontology for the Distribution of Difference between RDF Graphs. http://www.w3.org/DesignIssues/Diff, 2004

[3] M. Biezunski, M. Bryan, S. Newcomb, ISO/IEC 13250 TopicMaps

[4] G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, G. Antoniou, Ontology change: classfication and survey. The Knowledge Engineering Review, 23(2), 2008

[5] P. Hayes and B. McBride, RDF Semantics. Technical Report, W3C Recommendation, 2004

[6] G. Klyne, J. J. Carroll, B. McBride, Resource Description Framework(RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004

[7] D. L. McGuinness and F. V. Harmelen, OWL Web Ontology Language Overview, W3C Proposed Recommendation, 2003

[8] Natalya F. Noy and Michel Klein, Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information System, 2004

[9] Natalya F. Noy and Mark A. Musen. Ontology Versioning in an Ontology Management Framework. IEEE Intelligent Systems, 19(4), 2004

[10] Natalya F. Noy and Mark A. Musen. PromptDiff: A Fixed-Point Algorithm for Comparing Ontology Versions. In Proceedings of AAAI, 2002

[11] G. Tummarello, C. Morbidoni, R. Bachmann-Gmur, O. Erling, RDFSync: efficient remote synchronization of RDF models. In Proceedings of the ISWC, 2007

[12] M. Volkel and T. Groza, SemVersion: An RDF-based Ontology Versioning System. In Proceedings of ICWI, 2006

[13] D. Zeginis, Y. Tzitzikas, V. Christophides, On the foundation of Computing Deltas between RDF models, In Proceedings of ISWC, 2007