

온톨로지 개체의 삭제 알고리즘 설계

이순미^o, 나세리^{*}

^o경인여자대학교 영상방송정보과

^{*}한양여자대학교 컴퓨터정보과

e-mail: leesm@kic.ac.kr, slna@hywomn.ac.kr

Design of Delete_Algorithm for Ontology Object

Soon-Mi Lee^o, Sae-Lee Na^{*}

^oDept. of Video Broadcasting & Information, KyungIn Women's University

^{*}Dept. of Computer Information, HanYang Women's University

● 요약 ●

최근에 웹 온톨로지 언어를 사용한 웹 서비스와 애플리케이션들이 많이 개발됨에 따라 RDF/S를 기반으로 한 온톨로지 문서의 크기가 증가되었고 이로 인해 효율적으로 문서를 저장하고 검색하기 위한 시맨틱 웹 정보관리 시스템의 필요가 대두되었다. 본 논문에서는 시맨틱 웹 정보관리 시스템에서 클래스와 프로퍼티의 삭제 알고리즘을 제안한다. 제안한 삭제 알고리즘에서는 클래스나 프로퍼티가 삭제될 때에 이를 반영하여 계층 정보의 스키마 정보가 변경된다.

키워드: 시맨틱 웹(Semantic Web), 온톨로지(Ontology), RDF, RDFS, 계층 정보

I. 서론

시맨틱 웹에서는 웹 상의 정보에 의미를 부여함으로써 컴퓨터가 사용자를 대신하여 웹 정보자원의 의미를 이해하고, 정보의 검색, 추출, 해석, 재사용 등 제반 처리를 수행한다. 이러한 시맨틱 웹을 실현하기 위한 핵심적인 기술이 바로 온톨로지(Ontology)이다. 온톨로지 정보를 관계형 데이터베이스에 저장하고 검색하는 [1]의 연구에서는 특별히 계층 정보를 관계형 테이블에 저장하고 효과적으로 검색할 수 있는 저장 스키마를 제안하고 있다. 본 논문에서는 [1]에서 제안한 관계형 시맨틱 웹 정보 관리 시스템에서 개체가 삭제될 때 계층 정보를 자동으로 삭제 및 변경하는 알고리즘을 제안한다.

본문의 구성은 다음과 같다. 2장에서는 관련 연구를, 3장에서는 클래스나 프로퍼티가 삭제될 때 계층 정보를 나타내는 테이블의 내용이 삭제 및 변경되는 알고리즘을 제시하며 4장에서 결론을 내린다.

II. 관련 연구

온톨로지 문서들을 저장하고 관리하는 방법은 DLDB[2]의 논문에서 처음으로 논의된 이후 Sesame[3], Jena[4], RDFSuite[5] 등의 유사한 연구가 진행되었다. 온톨로지 정보를 관계형 데이터베이스에 저장하고 검색하는 [1]의 연구에서는 계층 정보의 처리를 위하여 별도의 XML 문서를 유지할 필요가 없이 계층정보를 검색하는 것이 가능했음을 뿐만 아니라 계층정보의 추출 시에 계

층 구조가 깊어질수록 하위 구조를 찾아가는 루틴이 계속 반복되어 효율성이 저하되는 문제를 개선하였다. 본 논문에서는 [1]의 연구를 확장하여 클래스나 프로퍼티(속성)가 삭제될 때에 이를 반영하여 계층 정보의 스키마가 삭제 및 변경되게 하는 알고리즘을 제안하고자 한다.

III. 삭제 알고리즘

1. 온톨로지 저장구조

본 논문에서는 온톨로지 문서에 선언된 클래스/프로퍼티에 대한 계층 정보를 나타내기 위하여 cHierarchy/ pHierarchy 테이블을 사용하였다. 클래스의 계층 정보를 저장하고 있는 테이블인 cHierarchy와 프로퍼티의 계층 정보를 저장하고 있는 테이블인 pHierarchy에 preorder와 postorder 필드를 두어 조상/자손(ancestor/ descendant)의 관계를 나타내도록 하였는데 cHierarchy와 pHierarchy의 구성은 <표1>과 같다. <표1>에서 superClass는 상위 클래스를 나타내주며 preorder와 postorder는 계층정보를 나타내주는 필드로서 전위, 후위 정보가 저장된다. 전위(Preorder)와 후위(Postorder) 숫자의 순서쌍을 대응시켜 계층 정보로 활용한다. 전위의 넘버링 방법은 상위 레벨에서 하위 레벨 순서로, 좌측에서 우측 순서로 번호가 부여되며 후위의 넘버링은 좌측 하위 노드에서 시작하여 좌에서 우로 하위에서 상위 레벨 순으로 번호가 부여된다.

cHierarchy 테이블을 사용하여 하위 클래스를 검색하는 방법은

superClass 값이 자신의 superClass 값과 같으며 preorder 값이 자신의 전위 값보다 크고 postorder 값이 자신의 후위 값보다 작은 클래스를 모두 찾으면 된다[1].

표 1. cHierarchy와 pHierarchy의 구성
Table 1. Component of cHierarchy and pHierarchy

cHierarchy				
clD	cName	superClass	preorder	postorder

pHierarchy				
pID	pName	superProp	preorder	postorder

2. 삭제 알고리즘

시맨틱 웹 정보 관리 시스템에서 클래스나 프로퍼티가 삭제되면 계층정보를 나타내는 cHierarchy 와 pHierarchy 테이블에 삭제 내용이 반영되어야 한다. cHierarchy 와 pHierarchy 테이블의 노드 삭제와 전위 및 후위 정보를 변경시키는 삭제 알고리즘은 다음의 두 가지 방법으로 처리될 수 있다.

2.1 삭제하려는 개체가 단말 노드인 경우

삭제하려는 노드 : x

- step1. x노드를 검색한다.
- step2. x의 preorder값보다 preorder가 큰 모든 노드의 preorder를 1씩 감소한다.
- step3. x의 postorder값보다 postorder가 큰 모든 노드의 postorder를 1씩 감소한다.
- step4. x를 제거한다.

2.2 삭제하려는 개체가 단말 노드가 아닌 경우

이 경우에는 삭제하려는 개체의 하위 개체들까지 모두 제거되는 연쇄 삭제(Triggered Deletion)가 발생되어야 한다.

- step1. x노드를 검색한다.
- step2. x의 모든 자손노드를 검색한다.
- step3. x와 x의 자손노드들 중 가장 큰 preorder값과 가장 작은 preorder값을 검색하여 각각 maxpre와 minpre로 지정한다.
- step4. x와 x의 자손노드들 중 가장 큰 postorder값과 가장 작은 postorder값을 검색하여 각각 maxpost와 minpost로 지정한다.
- step5. 모든 노드에 대해서 다음을 수행한다.

- 5.1 preorder>maxpre이면 preorder를 자손노드수+1만큼 감소시킨다.
- 5.2 minpre<preorder<maxpre이면 preorder를 (preorder가 작은 자손노드수+1)만큼 감소시킨다.
- 5.3 postorder>maxpost이면 postorder를 자손노드수+1만큼 감소시킨다.
- 5.4 minpost<postorder<maxpost이면 preorder 를 (postorder가 작은 자손노드수+1)만큼 감소시킨다.

step6. x와 x의 자손을 제거한다.

위 알고리즘에서 자손노드를 검색하는 방법은 [1]의 방법을 사용한다.

IV. 결론

본 논문에서는 시맨틱 웹 정보 관리 시스템에서 임의의 개체가 삭제될 때 온톨로지 스키마의 계층 정보를 변경시키는 알고리즘을 설계하였다. cHierarchy 와 pHierarchy 테이블의 preorder와 postorder 정보를 변경시키는 방법을 삭제하려는 개체가 단말 노드인 경우와 아닌 경우로 나누어서 설계하였다. 추후의 연구 과제로 본 논문에서 제안한 알고리즘을 구현하여 기존의 개체 삭제 방법과의 성능을 비교 분석하는 연구가 진행되어야 한다.

참고문헌

- [1] Soonmi Lee, "Design of Relational Storage Schema and Query Processing for Semantic Web Documents", Journal of The Korea Society of Computer and Information, Vol. 14, No. 1, pp. 35-45, 2009.
- [2] Z. X. Pan and J. Heflin, "DLDB: Extending relational databases to support semantic web queries", Workshop on Practical and Scalable Web Systems, pp. 109-113, 2003.
- [3] J. Broekstra, A. Kampman, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", Lecture Notes in Computer Science, Vol.2342, pp. 54-68, 2002.
- [4] K. Wilkinson, C. Sayers, H. Kuno, "Efficient RDF Storage and Retrieval in Jena2", SWAD, pp.131-150, 2003.
- [5] S. Alexaki, and et al., "The RDFSuite: Managing Voluminous RDF Description Bases", Technical Report, ICS_FORTH, 2001.