온톨로지 기반 보편적 DTD 생성기의 구현

공 용 해, 이 경 수
순천향대학교 정보기술공학부
전화 : 041-530-1320 / 핸드폰 : 011-245-2437

Implementation of Universal DTD Generator based on Ontology

Yong Hae Kong, Kyeung Soo Lee
Div. of Information Technology Engineering, Soonchunhyang University
E-mail : yhkong@sch.ac.kr, nio_lee@hotmail.com

Abstract

XML is widely used to provide information on the web. Since accessing XML documents depends on their structures, it is difficult to extract necessary information from documents having different structures. We constructed an ontology of a small application domain and implemented an universal DTD generator from the ontology. The universal DTD can be used in extracting necessary information from the various XML document types representing an identical idea.

Ⅰ. 서론

XML(eXtensible Markup Language)은 웹 상에서 새로운 표준 데이터로 받아 들여 지고 있다. XML은 기존의 HTML 문서에 비해 의미적인 태그(Tag)를 사용할 수 있기 때문에, 정보에 대한 표현과 검색을 보다 명확히 할 수 있다. XML의 태그는 문서에 따라 특정한 구조를 가지고 있으며, 이러한 문서 구조를 DTD로 정의한다. 따라서, 문서의 구조(의미적인 태그의 구조)를 정의한 DTD는 의미 정보를 포함하게 된다.

현재, 정보 검색을 위해 XML 문서에 접근하는 방법은 구조에 의존적이기 때문에 DTD에 내포된 의미 정보를 검색하지 못한다. 만약 모든 XML 문서가 같은 구조를 따른다면, 의미를 고려한 상태에서 동일한 방법으로 문서를 생성, 접근할 수 있다.

본 연구는 한국소프트웨어 진흥원의 ITRC 사업에 의해 수행된 것임.

그러나 대부분의 경우 태그의 의미나 문서 구조에 대한 별다른 동의 없이 XML 문서를 작성한다. 즉, 동일한 정보가 다양한 형태 구조로 표현 가능하므로, 각 문서마다 그 구조에 맞는 방법으로 접근해야 하는 현재의 정보검색 방법으로는 일관성 있는 정보검색에 어려움 있다.

본 논문에서는 서로 구조가 다른 XML 문서에 동일한 방법으로 접근하기 위해 온톨로지(Ontology)를 이용하여 XML 문서의 처리가 가능한 포괄적 DTD를 생성하는 DTD 생성기를 구현하였다. 그림 1은 다른 구조를 갖는 특정 응용 분야의 XML 문서에 온톨로지를 기반으로 하여 접근하는 과정을 도식화한 것이다.

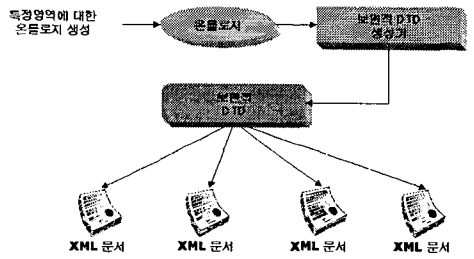

그림 1. 온톨로지 기반 XML 문서 접근 과정

Ⅱ. 온톨로지

온톨로지는 개념에 대한 명세서이다[1]. 온톨로지는 논리 언어의 표현법 중 일부를 이용하여 개념을 표현한다[2]. 본 논문에서는 'MusicCD' 분야에 대한 개념을 온톨로지 형식으로 표현하였다. 그림 2는 'MusicCD' 와 관련된 개념들을 계층구조를 표현한 것

이다. 예를 들어, 개념 Solo와 개념 Group은 개념 Singer의 하위 개념이라는 것을 의미하며, 상위 개념의 속성들은 하위 개념으로 상속된다.

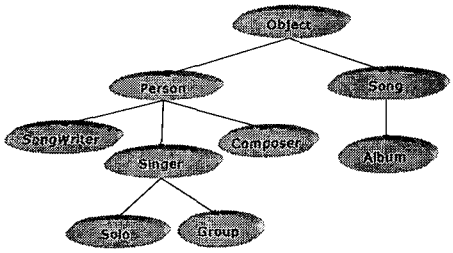

그림 2. 개념 계층구조

그림 3은 개념의 속성을 표현한 것으로 속성들은 STRING이나 다른 개념들과의 관계로 정의된다.

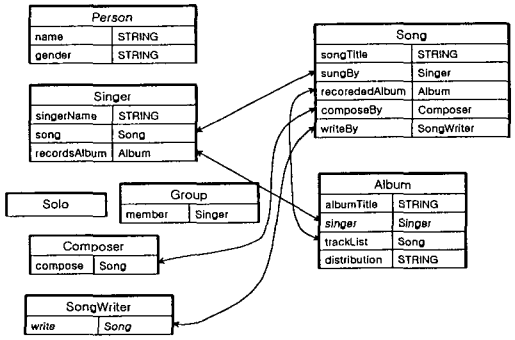

그림 3. 개념의 속성과 다른 개념들과의 관계

온톨로지는 개념의 계층구조와 다른 개념들과의 관계로 표현되며, 온톨로지를 이용하면 개념 레벨에 의한 의미 기반의 XML 문서 처리가 가능하다[3].

Ⅲ. 보편적 DTD 생성기

온톨로지에서 표현된 개념 레벨의 정보를 DTD로 표현하기 위해, DTD 생성기는 온톨로지에서 개념과 속성을 추출하고, 이를 DTD의 표현 방법에 맞게 다시 표현함으로써 보편적 DTD를 생성한다. 보편적 DTD 생성기는 두 개의 프로시져로 구성되어 있다.

첫 번째는 Ontology Processor로, 온톨로지를 파싱하여 온톨로지를 그대로 반영한 개념 클래스와 속성 클래스를 생성한다. 두 번째는 Generating DTD로, 개념 클래스와 속성 클래스를 이용하여 DTD 생성 방법에 의해 보편적 DTD를 생성한다. 그림 4는 보편적 DTD 생성기의 구조이다.

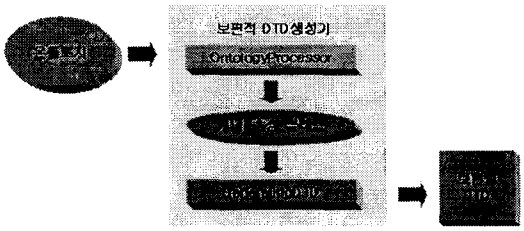

그림 4. 보편적 DTD 생성기의 구조

보편적 DTD 생성기는 온톨로지에 표현된 개념과 속성을 DTD 문법에 맞게 반영해야 한다. 이를 위하여 다음과 같은 4가지 과정을 거친다.

- 개념 구조를 위한 ENTITY 생성
- 개념과 그 속성에 대한 ELEMENT 생성
- 개념의 속성들에 대한 ATTLIST 생성
- 속성들에 대한 ELEMENT 생성

4.1 개념 구조를 위한 ENTITY 생성

온톨로지에 표현된 계층구조를 DTD의 매개변수 ENTITY로 표현하는 과정으로, 개념들 간의 상속을 적용시킨다. 어떠한 개념이 주어지면, 그 개념을 상위 개념으로 갖는 하위 개념을 재귀적으로 순회한다. 하위 개념을 찾으면, ENTITY로 설정한다. 표 1은 개념 구조를 위한 ENTITY 생성 방법이다.

표 1. 개념 구조를 위한 ENTITY 생성 방법

```
개념리스트의 끝까지 순회
    다음 개념으로 이동한 후 재귀호출
    자신의 하위개념이면
        다음 개념으로 이동한 후 재귀호출
```

개념 구조를 DTD에 반영하기 위해 % 기호를 매개변수 entity 참조로 사용한다. 개념 Singer는 온톨로지의 계층구조(상속)에 의해 Group을 가질 수 있다.

```
<!ENTITY % Singer "Singer | Solo | Group">
```

4.2 개념과 그 속성에 대한 ELEMENT 생성

자신의 속성을 정의하기 전에 상위 개념의 속성을 상속받아서 자신의 속성에 포함 시켜야 한다. 이를 위하여 재귀 호출로 상위 개념을 순회하여 상속을 받은

후에, 마지막으로 자신의 속성을 순회한다. 표 2는 개념과 그 속성에 대한 ELEMENT 생성 방법이다.

표 2. 개념과 속성에 대한 ELEMENT 생성 방법

```
개념리스트의 끝까지 순회
  상위 개념 리스트 순회
    재귀호출
  속성이 존재하면
    속성 리스트 순회
      속성 상속
```

개념의 속성을 DTD로 표현하기 위해 ELEMENT로 이용한다. 여기에서는 Mixed Element를 이용하여, subelement와 문자 데이터를 혼합한 값을 갖게 한다. 이는 관계된 어떠한 값이라도 모두 상속받게 하기 위해서이다. 목록의 처음에는 #PCDATA를 사용하고, 그 뒤에 subelement를 혼합시킨다. 예를 들면, 개념 Composer는 name과 gender 같은 속성을 상속받아서 3개의 속성을 포함하게 된다.

```
<!ELEMENT Composer (#PCDATA | name | gender | compose)* >
```

4.3 개념의 속성들에 대한 ATTLIST 생성

4.2의 개념과 그 속성에 대한 ELEMENT 생성과 같은 방법을 적용한다. 하위 개념이 존재하면, Object를 찾을 때까지 재귀적으로 순회하고, Object를 찾으면 되돌아오면서 속성들을 상속받는다. 온톨로지의 속성을 DTD의 속성으로 표현하기 위해 ATTLIST를 이용한다. 옵션으로 #IMPLIED를 이용하여 필요에 따라 사용되는 선택사항임을 표시한다.

```
<!ATTLIST Composer
        name            CDATA #IMPLIED
        gender          CDATA #IMPLIED
        compose         CDATA #IMPLIED>
```

4.4 속성들에 대한 ELEMENT 생성

각 개념들 간의 관계가 표현되는 과정으로, 개념의 속성들을 순회하면서 관계된 개념을 찾는다. 속성 값이 STRING이 아닌 개념일 때, 해당 개념에서 자신과 관계 있는 속성이 존재하는지 검색한다. 이 과정에서 개념들간의 관계가 맺어진다. 표 3은 속성에 대한 ELEMENT 생성 방법이다.

표 3. 속성에 대한 ELEMENT 생성 방법

```
개념리스트의 끝까지 순회
  속성이 존재하면
    속성리스트 순회
      속성값 STRING이면
        #PCDATA 출력
      속성값이 개념이면
        개념리스트 순회
          속성값과 개념값이 서로 존재하면
            해당 개념 출력
```

각 속성의 타입이 STRING인 것은 다른 개념들과 연관 관계가 없는 것이므로 #PCDATA로 표현한다. 다른 개념을 참조하는 것은 그 개념과 자신과의 연관 관계를 검색하고, 관계가 존재하면 #PCDATA 이후에 관계있는 매개변수 ENTITY 목록을 적어준다.

```
<!ELEMENT member(#PCDATA | %Singer;)* >
```

이러한 과정에 의하여 온톨로지의 개념과 속성, 그리고 그것들의 관계를 DTD의 문법에 맞게 표현하게 된다. 그 결과로, 온톨로지를 기반으로 한 보편적 DTD가 생성된다. 표 4는 DTD 생성기에 의해 생성된 보편적 DTD의 일부분이다.

표 4. 보편적 DTD의 일부분

```
<!-- Generated universal DTD base on the ontology -->
<!ENTITY % Person "Person | Composer | SongWriter | Singer | Group | Solo" >
<!ENTITY % Singer "Singer | Group | Solo" >
<!ENTITY % Group "Group" >
<!ENTITY % Song "Song | Album" >
<!ENTITY % Album "Album" >
                    :
<!ELEMENT Composer (#PCDATA | name | gender | compose)* >
<!ELEMENT Song (#PCDATA | songTitle | sungBy | writeBy | composeBy | recordedAlbum)* >
                    :
<!ATTLIST SongWriter
        name            CDATA   #IMPLIED
        gender          CDATA   #IMPLIED
        wrtite          CDATA   #IMPLIED
        composeByCDATA  #IMPLIED
        recordedAlbum   CDATA   #IMPLIED >
                    :
<!ELEMENT albumTitle (#PCDATA) >
<!ELEMENT distribution (#PCDATA) >
<!ELEMENT singer (#PCDATA | %Singer;)* >
<!ELEMENT trackList (#PCDATA | %Song;)* >
```

상기의 과정을 거쳐 DTD 생성기로부터 생성된 보편적 DTD는 다양한 XML 문서 구조를 포함하게 한다. 따라서, 온톨로지를 기반으로 생성된 보편적 DTD

는 동일한 방법으로 구조가 다른 XML 문서에 독립적으로 접근할 수 있다[4].

Ⅳ. 생성된 보편적 DTD의 XML 적용

XML 문서는 동일한 정보에 대해 서로 다른 구조로 표현할 수 있다. 다음의 XML 문서(MusicCD1.xml)는 앨범에 관한 것이다. 이 XML 문서는 Album을 루트 엘리먼트로 가지며, 앨범에 관한 내용을 element로 표현했다.

```
<Album>
  <albumTitle>Hell Freezes Over</albumTitle>
  <singer>Eagles</singer>
  <distribution>Universal Music</distribution>
  <trackList>
    <Song>
      <songTitle>Hotel California</songTitle>
      <writeBy>D.Felder, D.Henly</writeBy>
      <composeBy>G.Frey</composeBy>
    </Song>
  </trackList>
</Album>
```

다음 XML 문서(MusicCD2.xml)는 MusicCD1.xml과 같은 내용이다. 하지만, 앞에서 보인 문서와 다른 구조를 갖고 있다.

```
<Singer>
  <singerName>Eagles</singerName>
  <recordsAlbum>
    <Album albumTitle="Hell Freezes Over">
      <distribution>Universal Music</distribution>
      <trackList>
        <Song songTitle="Hotel California">
          <writeBy>D.Felder, D.Henly</writeBy>
          <composeBy>G.Frey</composeBy>
        </Song>
      </trackList>
    </Album>
  </recordsAlbum>
</Singer>
```

이와 같이, XML 문서는 같은 내용에 대하여 다른 구조를 갖기 때문에, DTD 또한 다르다. 하지만, 보편적 DTD는 두 가지 문서에 모두 적용 가능하다. 유효성(Validity)을 검사하기 위해 Microsoft에서 제공하는 'Internet Explorer Tools for Validating XML'을 이용하였다. 각각의 XML 문서에 서로 다른 DTD를 적용시키면 유효성 검사에 실패하지만, 보편적 DTD 생성기에 의해 생성된 보편적 DTD를 적용시킬 경우, 두 개의 XML 문서에 대하여 모두 'Validation Successful'이라는 결과를 얻게 된다. 즉, 생성된 DTD는 문서 구조에 독립적이기 때문에, 구조적으로 서로 다른 두 개의 XML 문서에 적용 가능하다.

Ⅴ. 결론

XML 문서는 의미적인 태그를 사용하지만 문서 접근 방법이 구조에 종속적이기 때문에. 일관성 있는 정보의 검색에 어려움이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 개념 레벨에서 문서에 접근하고자 온톨로지를 이용하였으며, 온톨로지를 기반으로 하여 보편적 DTD를 생성하는 DTD 생성기를 구현하였다.

보편적 DTD는 온톨로지를 따르는 다른 XML 문서에 모두 적용 가능하다. 이는 하나의 온톨로지와 DTD를 이용해서 다양한 XML 문서에 접근할 수 있다는 것을 의미한다. 즉, 구조적으로는 보편적 DTD에 의해서, 의미적으로는 온톨로지에 의해서 XML 문서를 처리할 수 있게 된다. 보편적 DTD는 온톨로지를 기반으로 생성되기 때문에 구조적인 DTD에 의미를 보완해 주게 되며, 이러한 보편성은 상호 관련 있는 XML 문서에 대한 규약을 정의하는 표준 DTD로 이용될 수 있다.

Reference

[1] T.R. Gruber: "A Translation Approach to Portable Ontology Specifications". in Knowledge Acquisition. vol. 6, no. 2, 1993. pp199-221.

[2] M. Kifer, G.Lausen, and J. Wu, "Logical foundations of object-oriented and frame-based languages". Journal of the ACM, 42, 1995.

[3] S.Decker, M. Erdmann, D. Fensel, and R. Studer, "Ontobroker: Ontology-based access to distributed and semi-structured information". R. Meersman et al. eds, Semnatic Issues in Multimedia Systems, Kluwer Academic Publisher, Boston 1999.

[4] M. Erdmann and R. Studer, "How to Structure and Access XML Document with Ontologies", Data & Knowledge Engineering, Vol. 36, No.3, pp. 317-335, 2001.