論文
3-7-1

개념정보를 포함한 포괄적 DTD 생성기의 설계 및 구현 (Design and Implementation of Concept Information Based Universal DTD Generator)

최 인 석* 공 용 해**
(In-Seok Choi) (Yong-Hae Kong)

요 약

인터넷상에는 다양한 정보 자원들이 존재하고 있으며 최근에는 XML에 대한 관심이 날로 증가하고 있다. XML은 정보의 구조를 자유롭게 정의할 수 있기 때문에 동일한 응용영역을 표현하는 문서들은 상호 이질적이므로 문서들의 특성에 맞는 각각의 응용프로그램이 필요하다. 만일, 특정 정보를 표현하는 XML문서에 적용되는 DTD가 개념정보를 포함하고 있다면 문서의 구조에 자유롭게 적용될 수 있으므로 상호 이질적인 문제를 극복할 수 있다.

본 연구에서는 개념정보가 포함된 포괄적 DTD를 자동으로 생성하기 위하여 DTD생성기를 개발하였다. 또한 응용영역을 개념화하여 포괄적 DTD를 생성하는데 사용하였다. 포괄적 DTD는 개념정보를 포함하고 있으므로 동일 응용영역 표현하는 다른 구조의 XML 문서에 모두 적용될 수 있으며, 문서의 유효성을 검증하고, 유연한 응용프로그램의 접근이 가능하게 하였다.

ABSTRACT

There are various information resources on the Internet and people are taking more interest in XML day by day. In XML, the structure of information can be freely defined so that the standardization of documents can be hardly made. If DTD, which is applied to an XML Document representing specific information, is including concept information, it can be freely applied to the structure of document and also contributes to the convenience in information retrieval.

In this study, we developed universal DTD Generator in order to automatically generate DTD including concept information. For the generation of universal DTD, the conceptualization of information is required; to conceptualize information, the hierarchical structuring and propertizing are required.

The hierarchical structuring represents the inclusive relation of routine concepts for representing information in hierarchical structure, and the propertizing represents the property and mutual relation that the each concept represented in hierarchical structure can have. The defined hierarchical structure and propertization come to generate the universal DTD Generator.

The universal DTD generated by DTD Generator can be applied to all the XML Documents representing the same information in different structure.

However, the most ideal way is that the information of universal DTD, which can be applied to various documents, is including all the cases. Therefore, the study for designing correct concept information is necessary.

* 정회원 : 홍성기능대학 전자계산기과 부교수
** 정회원 : 순천향대학교 정보기술공학부 교수
논문접수 : 2002. 5. 8.
심사완료 : 2002. 6. 3.
※ 본 연구는 한국소프트웨어 진흥원의 ITRC 사업에 의해 수행된 것임.

1. 서론

인터넷상에는 다양한 정보 자원들이 존재하고 있으며 최근에는 XML(eXtensible Markup Language)에 대한 관심이 날로 증가하고 있다[1]. XML은 매우 유연성 있는 텍스트형식으로서 정보의 구조를 자유롭게 정의할 수 있기 때문에 문서의 표준화가 어렵다[2]. 또한 XML문서를 작성할 때마다 문서의 구조를 정의해야 하는 불편함이 따르며, 동일한 정보에 대하여 서로 다른 구조와 속성을 지니고 있으므로 각각의 문서에 적합한 응용프로그램이 필요하다[3].

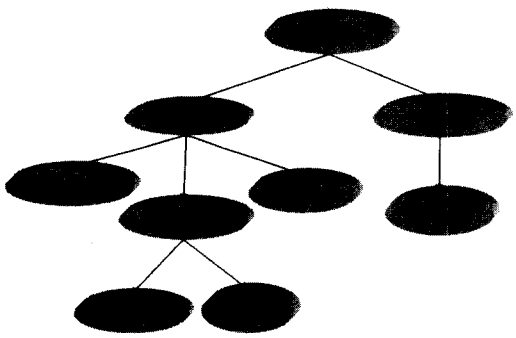
만일, 비정형 XML 문서들이 동일한 정보에 대해서 동일한 구조와 약속된 개념 표현을 따를 경우 문서를 작성할 때마다 문서의 구조를 정의해야 하는 어려움을 해결할 수 있지만, 현실적으로는 그렇지 못하다. 현재 W3C 등에서 XML 문서의 구조를 정의하는 DTD(Document Type Definition)의 표준화 작업을 진행하고 있으나 표준화된 DTD가 보편화되기까지는 많은 시간과 노력이 필요하며, 수많은 정보에 대해 모든 DTD를 표준화하기는 사실상 어려운 일이다[4]. 만약 특정 정보가 적절히 개념화되어 표현된다면, 이러한 개념 정보를 이용하여 비정형 XML 문서에 포괄적으로 적용할 수 있는 포괄적 DTD를 생성할 수 있다[5]. 포괄적 DTD는 동일한 응용영역을 표현하는 비정형 XML문서에 제약을 받지 않고 적용할 수 있으며, 문서의 유효성을 검증할 수 있다. 또한 동일한 응용영역의 정보를 표현하는 다양한 XML문서에 동일한 응용프로그램을 적용할 수 있게 한다. 본 연구는 포괄적 DTD를 생성하기 위하여 응용 영역을 개념화하고, DTD생성기를 개발한다.

2. 정보의 개념화

개념 수준의 정보는 응용영역의 정보 표현을 개념화하여 사용자와 응용 시스템들 사이에 전달될 수 있는 정보 영역의 공유 개념과 공통된 지식을 제공할 수 있게 하며, 웹 응용프로그램들 사이의 지식을 공유하고, 일관된 처리를 가능하게 한다. 이러한 개념수준의 정보는 지식의 공용과 재사용을 촉진하기 위하여 인공 지능 분야에서 사용되고 있다[6]. 정보를 개념화하기 위해서는 계층적 구조화와 속성화가 필요하다[7].

2.1 개념의 계층적 구조화

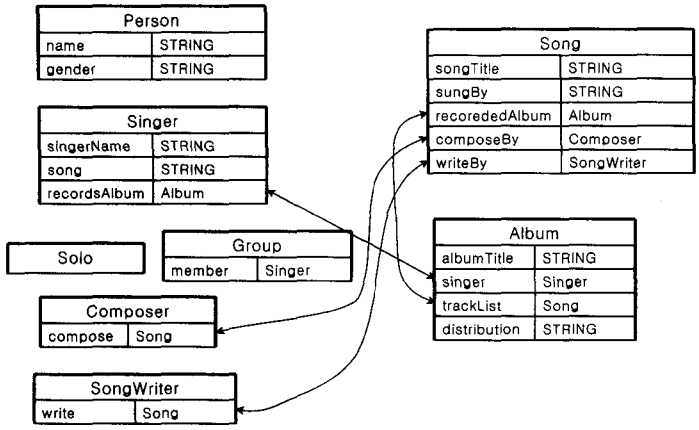
계층적 구조화는 정보를 표현하기 위한 응용영역 개념의 포함관계를 계층적 구조로 표현한다. [그림 1]은 음악 CD를 표현하는 계층적 구조의 예이다. 계층적 구조의 하위개념은 상위개념의 모든 속성을 상속받는다. 개념 'Person'은 하위 개념으로, 'songWriter', 'Singer', 'Composer', 'Solo', 'Group'을 가지며 하위개념은 상위개념 'Person'의 모든 속성을 상속받는다. 또한 개념 'Solo'는 상위개념 'Object'와 'person'의 속성을 상속받은 'Singer'의 모든 속성을 상속받는다. 정의된 개념적 계층구조는 DTD생성기에 의해서 적절한 포괄적 DTD를 표현하는데 사용된다.



[그림 1] 상품정보의 개념적 계층구조의 예

2.2 개념의 속성화

속성화는 계층적 구조로 표현된 각 개념들이 가질 수 있는 속성과 상호관계를 표현한다. 그림 2의 예는 정의된 개념들 사이의 속성과 상호 관계를 정의하였다. 개념 'Album'은 'singer'를 비롯한 여러 개의 에트리뷰트(attribute)와 그에 대한 값으로 정의되어 있다. 에트리뷰트 'singer'는 'Album'의 에트리뷰트로서 개념 'Singer'를 값으로 갖는다. 정의된 속성화는 DTD에서 표현되는 에트리뷰트들의 속성 값을 정의할 수 있게 한다.



[그림 2] 정의된 개념의 속성과 상호관계

3. 개념정보로부터의 DTD 생성

인터넷에 넓게 분포되어 정보를 표현하는 비정형 XML문서는 정보 표현과 구조가 다양하여 접근에 어려움이 따를 수 있다. 그러나 응용영역을 개념화한 정보로부터 유추된 DTD는 구조적 제약을 극복할 수 있으며 정보의 검색에 편리함을 제공할 수 있다..

3.1 DTD 생성기

구현된 DTD 생성기는 응용영역의 개념적 계층구조와 속성관계로부터 포괄적 DTD를 생성하기 위하여 몇 가지 과정을 거치게 된다.

3.2 개념 구조를 위한 ENTITY 생성

특정 개념을 상위개념으로 갖는 하위개념을 재귀적으로 순회한다. 개념의 순회는 최상위 개념인 'Object'를 만날 때까지 반복하면서 자신과 상속 관계에 있는 개념을 찾으면, ENTITY로 설정한다. 개념 'Singer'는 개념정보의 계층 구조에 의해 'Group'을 가질 수 있다. 따라서, 'Song'의 속성 'sungBy'는 'Singer'에 의한 상속된 개념을 가질 수 있다. [그림 3]은 DTD의 엔티티로 표현된 개념구조이다.

```
<!ENTITY % Object "Object | Song | Album |
Person | Composer | SongWriter | Singer |
Group | Solo" >
```

[그림 3] DTD ENTITY로 표현된 개념구조

3.3 개념과 그 속성에 대한 ELEMENT 생성

특정 개념의 속성을 정의하기 전에, 상위개념의 속성을 상속받아서 자신의 속성에 포함 시킨다. 이를 위하여 재귀 호출로 상위개념을 순회하여 상속을 받은 후에, 마지막으로 자신의 속성을 순회한다.

개념의 속성을 DTD로 표현하기 위해 ELEMENT를 이용하며, 개념과 관계된 어떠한 값이라도 모두 상속받게 하기 위해서 하위엘리먼트와 문자 데이터를 혼합한 값을 갖게 한다. 목록의 처음에는 #PCDATA를 사용하고, 그 뒤에 하위엘리먼트를 혼합시킨다. 예를 들면, 개념 'Composer'는 속성 'name'과 'gender'를 상속받아서 3개의 속성을 포함하게 된다. [그림 4]에서는 DTD ELEMENT로 표현된 개념 속성의 일부를 보여준다.

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

[그림 4] DTD ELEMENT로 표현된 개념의 속성

3.4 개념의 속성들에 대한 ATTLIST 생성

개념정보의 속성을 DTD의 속성으로 표현하기 위해 ATTLIST를 이용한다. 옵션으로 #IMPLIED를 이용하여 필요에 따라 사용되는 선택사항임을 표시한다. 이를 위하여 재귀 호출로 상위개념을 순회하여 상속을 받은 후에, 마지막으로 자신의 속성을 순회한다. 하위개념이 존재하면 'Object'를 찾을 때까지 재귀적으로 순회하고, 'Object'를 찾으면 되돌아오면서 속성들을 상속받는다. [그림 5]는 상위개념으로부터 속성을 상속받은 ATTLIST를 보여준다.

```
<!ATTLIST Singer
  name      CDATA #IMPLIED
  gender    CDATA #IMPLIED
  song    CDATA #IMPLIED >
```

[그림 5] 상위개념의 속성을 상속받은 ATTLIST

3.5 속성들에 대한 ELEMENT 생성

각 개념들 간의 관계가 표현되는 과정으로, 개념의 속성들을 순회하면서 관계된 개념을 찾는다. 각 속성의 타입이 STRING인 것은 다른 개념들과 연관관계가 없는 것이므로 #PCDATA로 표현한다. 다른 개념을 참조하는 것은 그 개념과 자신과의 연관 관계를 검색하고, 관계가 존재하면 #PCDATA 이후에 관계있는 개념 목록을 적어준다. [그림 6]은 생성된 DTD엘리먼트의 일부이다.

```
<!ELEMENT singerName (#PCDATA) >
<!ELEMENT recordsAlbum (#PCDATA | %Album;)* >
```

[그림 6] 생성된 DTD ELEMENT

4. 생성된 포괄적 DTD의 XML 적용

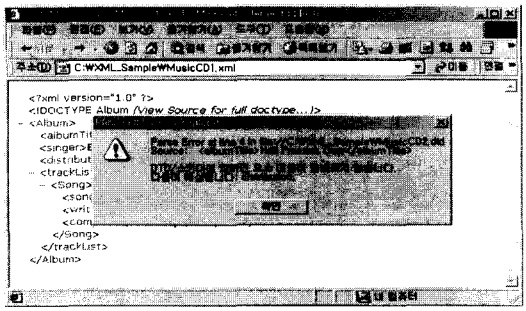
XML 문서는 동일한 상품정보에 대해 서로 다른 구조로 표현할 수 있다. 다음의 XML 문서(MusicCD1.xml)는 앨범상품에 관한 내용으로 구성되어 있다. 이 XML 문서는 Album을 루트 엘리먼트로 가지며, 앨범에 관한 내용을 엘리먼트로 표현했다.

```
<Album>
  <albumTitle>Hell Freezes Over</albumTitle>
  <singer>Eagles</singer>
  <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 문서는 Album을 기준으로 내용을 표현하였고, MusicdCD2.xml은 Singer를 기준으로 내용을 표현하였다. 즉, 두 개의 XML 문서는 서로 다른 구조를 갖고 있다.

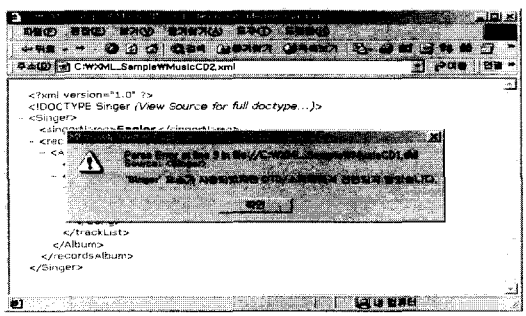
```
<Singer>
  <singerName>Eagles</singerName>
  <recordsAlbum>
    <Album albumTitle="Hell Freezes Over">
      <trackList>
        <Song songTitle="Hotel California">
          <writeBy>D.Felder, D.Henly</writeBy>
          <composeBy>G.Frey</composeBy>
        </Song>
      </trackList>
    </Album>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

이와 같이, XML 문서는 같은 내용에 대하여 다른 구조를 갖기 때문에, DTD 또한 다르다. 따라서, MusicCD1.dtd는 MusicCD1.xml에만 적용 가능하고 MusicCD2.dtd는 MusicCD2.xml에만 적용 가능하다. 하지만, 포괄적 DTD는 두 가지 문서에 모두 적용 가능하며, 그 유효성(Validity)을 검사하기 위해 Microsoft에서 제공하는 'Internet Explorer Tools for Validating XML'을 이용하였다.



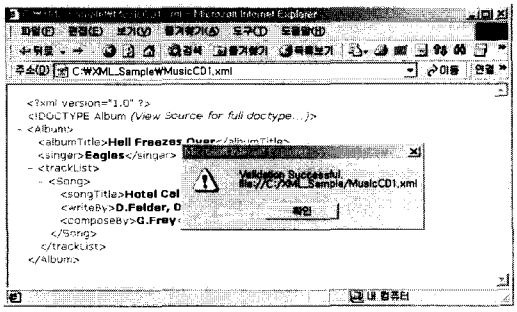
[그림 7] MusicCD1.xml에 MusicCD2.dtd를 적용시킨 예

[그림 7]은 MusicCD1.xml에 MusicCD2.dtd를 적용시킨 결과이다. 구조가 다른 DTD를 적용시켰기 때문에 유효성 검사에 실패하게 된다. 마찬가지로 MusicCD2.xml에 MusicCD1.dtd를 적용시키면 이 역시 구조가 맞지 않는 DTD를 적용시켰기 때문에 유효성 검사에 실패하게 된다. [그림 8]은 MusicCD2.xml에 MusicCD1.dtd를 적용시킨 후 유효성을 검사한 결과이다.



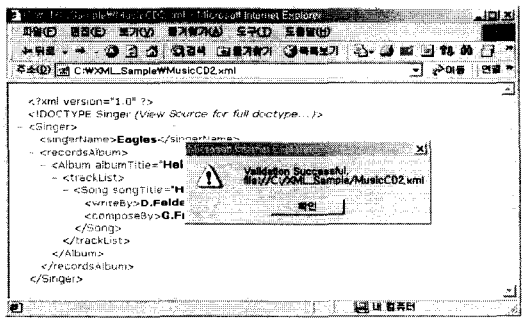
[그림 8] MusicCD2.xml에 MusicCD1.dtd를 적용시킨 경우

이처럼 XML과 DTD는 구조에 의존적이기 때문에, 같은 내용을 표현한 다른 XML에 조금이라도 다른 DTD를 적용시키면 유효성 검사에 실패하게 된다. 하지만, 포괄적 DTD는 두 개의 문서 - MusicCD1.xml과 MusicCD2.xml - 에 모두 적용 가능하다. [그림 9]는 MusicCD1.xml에 포괄적 DTD를 적용시킨 후 유효성을 체크한 결과 화면이다.



[그림 9] 포괄적 DTD에 의한 MusicCD1.xml 유효성 검사

마찬가지로 MusicCD2.xml에도 포괄적 DTD 생성기에 의해 생성된 포괄적 DTD를 적용시킨 후, 유효성을 체크하면 [그림 10]와 같은 결과를 얻을 수 있다.



[그림 10] 포괄적 DTD에 의한 MusicCD2.xml 유효성 검사

위와 같이 각각의 XML 문서에 서로 다른 DTD를 적용시키면 유효성 검사에 실패하지만, 포괄적 DTD 생성기에 의해 생성된 포괄적 DTD를 적용시킬 경우, 두 개의 XML 문서에 대하여 모두 'Validation Successful'이라는 결과를 얻게 된다. 포괄적 DTD 생성기에 의해 생성된 포괄적 DTD는 문서 구조에 독립적이기 때문에, 구조적으로 서로 다른 두 개의 XML 문서에 적용 가능하다.

5. 결론

최근 인터넷에는 XML 문서를 기반으로 하는 많은 정보 자원들이 제공되고 있다. XML은 정보의 구조를 자유롭게 정의할 수 있기 때문에 문서의 표준화가 어렵다. 따라서 많은 비정형 XML 문서들은 동일한 정보에 대하여 서로 다른 구조와 속성을 지니고 있으므로 응용프로그램의 접근이 극히 제한적일 수 있다. 만일, 동일 응용영역을 표현하는 XML 문서들이 동일한 구조와 약속된 표현을 따를 경우 문서를 작성할 때마다 문서의 구조를 정의해야 하는 어려움을 해결할 수 있으며 응용프로그램의 접근이 자유로울 수 있으나, 문서 작성에 많은 제약이 따르게 된다.

본 연구는 이러한 문제들을 해결하고 융통성 있는 XML 정보 문서의 구축과 접근을 위하여 응용영역을 개념화하고, 개념 정보를 내포한 포괄적 DTD를 자동으로 생성해주는 DTD생성기를 개발하였다. 포괄적 DTD는 유사한 정보를 표현하는 다양한 비정형 XML문서에 범용적으로 적용 가능하여 응용영역의 XML문서를 작성할 때마다 문서의 구조를 정의해야 하는 문제를 극복할 수 있었다. 또한 동일한 응용영역을 표현하는 XML문서의 유효성 검증과 응용프로그램의 유연한 접근이 가능하였다.

※ 참고문헌

[1] http://aicore.chungbuk.ac.kr/~dobest/document/Agents/AmEC/amec.html.

[2] Richard Anderson외, Professional XML, Wrox Press, 2000.

[3] Kristin Stock, David Pullar, "Identifying Semantically Similar Elements in Heterogeneous Spatial Databases Using Predicate Logic Expressions", Lecture notes in computer science, no. 1580, pp. 231-252, 1999.

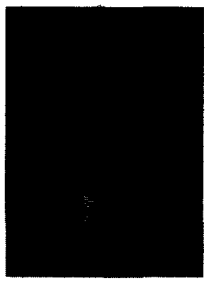
[4] W3C, Extensible Markup Language(XML) 1.0 Specification, http://www.w3.org/TR/ REC-xml, 1998.

[5] Michael Erdmann, Stefan Decker, "Ontology-aware XML-Queries"

[6] A. Gomez-Perez and R. Benjamins "Applications of Ontologies and Problem Solving Methods," AI Magazine, Vol. 20, No. 1, pp. 199-222, 1999.

[7] Sokratis Karkalas, Nigel Martin, "Automatic Semantic Object Discovery and Mapping from Non-nomalised Relation Databased Systems, ADVIS2000, pp. 92-107, 2000.

최인석



1988년 서울산업대학교 전산학과 학사
1996년 서울산업대학교 전산학 석사
2000년 순천향대학교 전산학 박사수료
1992년 한국산업인력공단
1995년 ~ 현재 홍성기능대학 전자계산기과 부교수
관심분야 : 이동에이전트, XML, 인터넷 응키워드 : 개념화, 개념 정보, 포괄적DTD, XML

공용해

1982년 연세대학교 전자공학과 학사
1986년 Polytechnic University 전산학 석사
1991년 Polytechnic University 전산학 박사
1982년 한진중공업 연구원
1983년 삼성전자 연구원
1991년 ~ 현재 순천향대학교 정보기술공학부 교수
관심분야 : 지능에이전트, 신경회로망, 컴퓨터비전 등