제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

유사구조를 갖는 XML 문서의 재구성을 위한 점진적인 시스템 설계

설진안*, 정계동*, 최영근*
*광운대학교 컴퓨터과학과
e-mail: {nicolas, gdchung, ygchoi}@kw.ac.kr

Design of an Incremental System for Reconstruction of Similar Structured XML Documents

Jin-An Seol*, Kye-Dong Jung*, Young-Keun Choi*
*Dept. of Computer Science, Kwang-Woon University

요 약

XML은 통합된 데이터 모델을 지원하기 위한 언어로, 특정 분야의 데이터에 대한 교환 및 통합의 필요성이 증대되어지고 있다. 일반적으로 데이터 교환은 다양한 공급자에 의해 독립적으로 운용 및 서비스됨으로서 개별적으로 데이터를 수집해야 하며 재배포 과정 또한 어렵다. 따라서 데이터 재배포 과정을 간소화하고 데이터 교환의 최적화을 위해 데이터 통합을 위한 재구성 방법이 필요하다. 본 논문에서는 특정 분야의 유사한 구조로 구성된 여러 문서를 입력받아 하나의 통합된 문서로 재구성할 수 있는 시스템을 제안한다. 제안된 시스템은 색인기법을 기반으로 추출된 정보를 하나의 문서로 매핑하기 위해 데이터 사전을 설계하고, 하나의 통합된 문서를 점진적인 과정을 통하여 재구성한다. 따라서 재구성된 문서는 재배포 과정을 간소화할 수 있으며, 데이터 교환의 최적화는 물론 전자문서교환(EDI)에 있어서 정보교환 능력을 증가시킬 수 있다.

1. 서론

XML은 W3C에서 발표한 산업표준으로서 데이터교환 모델로서 XML의 장점은 통합된 데이터 모델을 지원하기 위한 최상의 언어이다[1][2]. XML은 운영체제, 프로그램언어에 대해 독립적으로 데이터 교환이 가능하며, 대부분의 중요한 데이터베이스 벤더와 기업용 어플리케이션들이 XML을 지원하고 있으며, 지원 범위는 더욱 확대되고 있다[7].

XML은 정보의 표현과 재활용을 쉽게 할 수 있는 환경을 제공 하므로서 XML의 여러 가지 장점들을 이용해 다양한 응용분야에 대한 연구가 국내외에서 활발히 수행되어지고 있다. 그러나 특정 분야 정보에 대한 교환 및 통합의 필요성이 증대되고 있으나 이러한 정보들은 다양한 공급자에 의해 독립적으로 운용 및 서비스되고 있다.

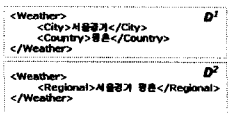
본 논문에서는 이러한 특정 분야(여행)에서 다양한 공급자에 의해 독립적으로 제공되는 정보를 입력받아 하나의 통합된 문서로 재구성하여 사용자의 편의에 맞게 다양한 포맷으로 서비스하기 위한 시스템을 설계 및 구현한다. 2장에는 이와 관련된 연구에 대해 기술하고, 3장에는 색인기법을 이용하여 추출된 정보를 Lore기법으로 작성한 최대 스키마 문서와 매핑하기 위해 데이터 사전을 설계하고, 하나의 통합된 문서를 점진적인 과정을 통하여 재구성하는 방법에 대해 기술한다. 마지막으로 4장에서는 결론 및 향후 연구과제에 대해 기술한다.

2. 관련연구

XML 문서를 이용하여 서로 다른 시스템이나 어플리케이션들이 데이터 교환의 신뢰성과 정확성을 제공하기 위해서는 문서의 구조적 정보가 중요하다. DTD(Document Type Definition)는 XML 문서를 구성하는데 필요한 엘리먼트, 속성, 엔티디 등 각 엘리먼트간의 구조적 관계를 정의한다. 하지만 DTD는 확장성을 갖지 못하는 단점을 갖고 있다. DTD를 갱신 했을 경우, 이전의 검증된 모든 문서들을 다시 검증해야하므로 DTD를 변경하지 않고서는 XML 문서를 확장시 킬 수 없다[6].

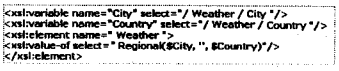
따라서 이 장에서는 XML 문서에 대한 DTD 문서의 구조정보를 이용하여 변환 및 병합하거나 저장 및 검색에 관련해서 진행되어온 연구에 대해 살펴본다. 구조적인 문서를 변환하거나 병합하기 위한 방법으로는 다음과 같다.

첫 번째, XSLT(Extensible Style Language Transformation)는 XML 문서를 마크업 언어로 재구성할 수 있는 기능을 제공한다. 예를 들면, 컨텐츠 정보 동일하지만 마크업이 다른 경우 반대로 마크업은 동일하지만 컨텐츠가 다를 경우 XSLT를 이용하면 쉽게 두 문서를 하나의 문서로 통합할 수 있다[3]. [그림1]은 두 문서에서 같은 컨텐츠를 제공하지만 마크업의 구조가 다를 경우가 메타정보를 포함하고 XML 문서를 표시하고 있다.



[그림1] 동일한 정보 포함하고 있지만 상이한 계층구조 갖는 두 문서

[그림2]는 D^2를 기준으로 D^1을 D^2로 통합하기 위한 XSLT 문장을 표시하고 있다.



[그림2] XSLT를 이용한 문서 병합

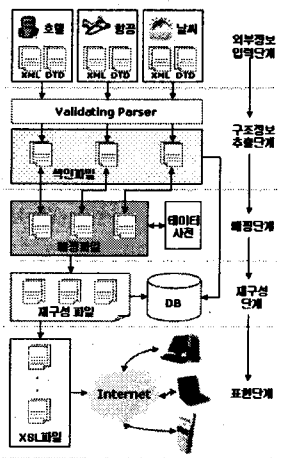
두 번째, 효율적으로 구조정보를 검색하여 데이터 저장을 위한 색인 기법으로는 K-ary 완전트리 기법, ID Assignment기법 등이 있다[4]. K-ary 완전 트리(K-ary Complete Tree) 색인 기법은 SGML 문서를 K-ary 완전 트리 매핑 과정을 통해 구조검색을 지원하기 위한 색인기법으로 부모노드와 자식노드의 관계를 간단한 수식을 이용하여 빠르게 검색할 수 있 장점을 갖고 있지만 가상 노드까지 노드 번호를 부여하는 단점을 가지고 있으며 조상/부모/형제/자손에 대한 계층정보와 순서 정보를 알기 어렵다. ID Assignment 색인 기법은 구조화된 문서의 문법적인 구조를 부모와 자식의 관계를 색인 내에서의 경로 표현을 구하기 위해 문서 구조의 특정 형태를 취하는 추상화에 기반한 색인 기법으로 자신의 노드 ID 앞에 부모의 노드를 추가하는 방법으로 자동적으로 자신의 조상노드의 정보를 알 수 있음은 물론 형제 노드에 대해서는 순서를 부여 하므로서 조상/부모/형제/자손에 대한 계층정보와 순서 정보를 간단하게 알 수 있다[4].

본 논문에서는 색인기법을 기반으로 데이터 사전을 이용하여 두 문서의 매핑과정을 점진적인 과정을 통하여 재구성하는 기법에 대해 기술한다.

3. 점진적인 재구성 시스템 설계

여러 개의 관련된 문서를 하나의 문서로 재구성하여 서비스하기 위한 적절한 적용 예를 여행에 필요한 여러 가지 정보를 하나의 문서로 재구성하는 방법에 대해 기술한다. 여행자가 여행에 필요한 정보는 항공권정보, 숙박정보, 날씨정보로 크게 3가지로 압축될 수 있다. 이러한 정보는 색인기법을 기반으로 데이터 사전을 이용하여 두 문서의 매핑 과정을 거쳐, 하나의 통합된 문서로 재구성되는 시스템은 다음과 같은 점진적인 과정을 거친게 된다.

첫 번째, 해당 XML 문서와 DTD 문서의 유효성(Validating) 검증과정, 두 번째, 문서의 계층구조와 엘리먼트들 간의 관계정보를 색인 파일로 작성하여, 구조정보 색인파일을 생성하는 구조정보 추출부분, 세 번째, 생성된 색인파일간의 매핑작업을 위해 메타사전을 이용한 매핑부분, 네 번째, 매핑된 정보를 이용하여 하나의 문서로 재구성하여 기 위한 재구성 부분, 다섯 번째, 재구성된 문서를 사용자의 편의에 맞게 다양한 포맷으로 서비스하기 위한 표현부분으로 구성되어져 있으며 [그림3]은 전반적인 시스템의 구조를 표시하고 있다.



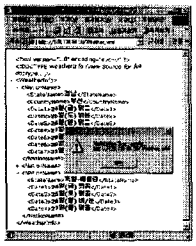
[그림3]시스템의 구조

3.1. 외부정보 입력 단계

여행자에게는 항공정보, 호텔정보, 날씨정보, 자동차 렌탈 정보 등을 취합하여 여행에 필요한 종합된 정보가 필요하다. 본 논문에서는 항공정보, 호텔정보, 날씨정보에 대한 XML 문서와 DTD 문서를 입력받아 하나의 종합된 정보를 제공하기 위해 문서의 고유번호(DID: Document ID)를 부여한다.

3.2. DTD 문서의 구조정보 추출

입력된 항공정보, 호텔정보, 날씨정보에 대한 XML 문서가 DTD 문서의 정의대로 올바르게 작성되었는지 유효성 검증 파서를 이용하여 유효성을 검사를 수행한다. XML 문서에서 엘리먼트의 자식엘리먼트 포함관계를 정의한 DTD 문서를 색인기법을 기반으로 루트 엘리먼트를 시작으로 터미널 엘리먼트 순으로 색인파일이 작성한다. [그림4]는 외부에서 입력된 날씨정보의 DTD 문서와 XML 문서를 나타내고 있다.



```
<!ELEMENT WeatherInfo (NationName)+>
<!ELEMENT NationName (StateName,
    CountryName?, Date1, Date2, Date3,
    Date4, Date5?, Date6?, Date7?)>
<!ELEMENT StateName (#PCDATA)>
<!ELEMENT CountryName (#PCDATA)>
<!ELEMENT Date1 (#PCDATA)>
<!ELEMENT Date2 (#PCDATA)>
<!ELEMENT Date3 (#PCDATA)>
<!ELEMENT Date4 (#PCDATA)>
<!ELEMENT Date5 (#PCDATA)>
<!ELEMENT Date6 (#PCDATA)>
<!ELEMENT Date7 (#PCDATA)>
```

[그림4] 외부에서 입력된 날씨정보의 DTD 문서와 XML 문서

입력된 날씨정보 DTD 문서에서 EID(Element ID)를 할당하고 추가적으로 EName(Element Name), ECS(Element ContentSpec), OType(Occur Type)정보를 추출하면 [그림5]와 같다. 각 엘리먼트에 EID를 할당하는 방법은 자신의 ID 앞에 부모의 ID를 추가하며 자신의 부모 및 조상 노드의 계층정보를 알 수 있음은 물론 같은 레벨에 있는 형제(Sibling) 노드의 경우 ID에 순서를 지정함으로서 형제(Sibling) 노드 간의 순서정보를 알 수 있다.

EID	PEID	EName	ECS	OType	ATID
1	0	WeatherInfo	Root	Null	
1	01	NationName	Element	+	
1	011	StateName	(#PCDATA)	Null	
1	012	CountryName	(#PCDATA)	?	
1	013	Date1	(#PCDATA)	Null	
1	014	Date2	(#PCDATA)	Null	
1	015	Date3	(#PCDATA)	Null	
1	016	Date4	(#PCDATA)	Null	
1	017	Date5	(#PCDATA)	?	
1	018	Date6	(#PCDATA)	?	
1	019	Date7	(#PCDATA)	?	

[그림5] 외부 날씨정보 DTD 문서의 엘리먼트 색인정보 (DTD_1)

[그림5]에서 ECS와 OType정보를 추출하는 이유는 다음 두 가지 이유이다. 첫 번째, ECS는 XML 문서에서 엘리먼트의 데이터 타입이 어떤 종류인지 나타내기 위함이다. 즉, 자식노드를 포함하는지 아니면 Leaf 노드인지 구분하기 위해서이다. 두 번째, OType는 XML 문서에서 모든 엘리먼트(자식 엘리먼트 포함)들의 출현 빈도수를 지정하기 위해서이다. 같은 방법으로 외부 DTD 문서와 패핑을 위한 표준 DTD 문서와 구조정보는 [그림6]과 같다.

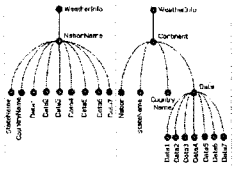
```
<!ELEMENT WeatherInfo (Continent+)>
<!ELEMENT Continent (Nation,
    StateName, CountryName?, Date)>
<!ELEMENT Date (Date1, Date2, Date3,
    Date4, Date5?, Date6?, Date7?)>
<!ELEMENT Nation (#PCDATA)>
<!ELEMENT StateName (#PCDATA)>
<!ELEMENT CountryName (#PCDATA)>
<!ELEMENT Date1 (#PCDATA)>
<!ELEMENT Date2 (#PCDATA)>
<!ELEMENT Date3 (#PCDATA)>
<!ELEMENT Date4 (#PCDATA)>
<!ELEMENT Date5 (#PCDATA)>
<!ELEMENT Date6 (#PCDATA)>
<!ELEMENT Date7 (#PCDATA)>
```

EID	EName	ECS	OType	ATID	ANO
0	WeatherInfo	Root	Null		
01	Continent	Element	+		
011	Nation	Element	Null		
012	StateName	Element	Null		
013	CountryName	(#PCDATA)	?		
014	Date	(#PCDATA)	Null		
0141	Date1	(#PCDATA)	Null		
0142	Date2	(#PCDATA)	Null		
0143	Date3	(#PCDATA)	Null		
0144	Date4	(#PCDATA)	Null		
0145	Date5	(#PCDATA)	?		
0146	Date6	(#PCDATA)	?		
0147	Date7	(#PCDATA)	?		

[그림6] 표준 날씨정보 DTD 문서의 엘리먼트 색인정보 (DTD_2)

그러나, DTD_2 문서와 DTD_1 은 같은 정보를 제공하지만 구조정보가 다음과 같이 다른 것을 알 수 있다. 첫 번째, DTD_2 문서에서 Date1~Date7 엘리먼트는 Date 라는 부모 엘리먼트를 갖지만, DTD_1 문서에서 Date1~Date7 엘리먼트는 NationName 이라는 부모 엘리먼트를 갖는다. 두 번째, 같은 정보를 제공하지만 엘리먼트의 이름이 서로 다른 엘리먼트가 존재한다. 세 번째, 루트 엘리먼트의 의미가 다르다.

이러한 이유는 표준 DTD는 작성에 필요한 데이터 및 구조정보를 여러 사이트에서 검토한 결과 날짜에 대한 데이터가 반복적으로 표시되지만 날짜 데이터 항목(개수)이 고정적이지 않았다. 오늘 기준으로 국내 날씨의 경우 7일 동안의 날씨를 제공하는데 비해, 국외 날씨의 경우 4일 동안의 날씨만 제공한다. 따라서 데이터의 수(날짜)가 고정적이지 않고 반복적이므로 날짜에 대한 데이터를 그룹화해야 할 필요가 있다. [그림7]은 날씨와 관련된 외부 DTD 문서와 공통적인 구조정보를 갖고 있는 표준 DTD 문서의 구조정보를 추출하기 위하여 반 구조적인 데이터 추출 기법인 최대/최소 경계스키마 추출 기법으로 두 문서의 스키마 그래프를 표시하고 있다.[5]



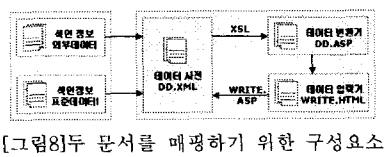
[그림7] DTD_1 과 DTD_2 문서의 스키마 그래프

따라서 다음 단계는 이처럼 같은 정보를 제공하지만 구조적으로 상이한 두 문서를 추출된 색인파일을 이용하여 매핑하는 과정에 대해 기술한다.

3.3. 색인정보를 이용한 두 DTD 문서의 매핑

같은 정보를 제공하는 두 문서를 하나의 문서로 재구성하기 위한 과정에서 여러 가지 문제점이 발생한다. 예를 들면, 내용은 같지만 구조가 상이하거나 엘리먼트의 이름만 다를 경우, 한쪽에는 있지만 다른 한쪽에는 없는 경우 등 여러 가지 문제점이 발생할 수 있다. 이 절에서는 2장에서 작성된 색인파일을 이용하여 외부 DTD 문서와 표준 DTD 문서를 재구성하기 위한 매핑방법에 대해 기술한다.

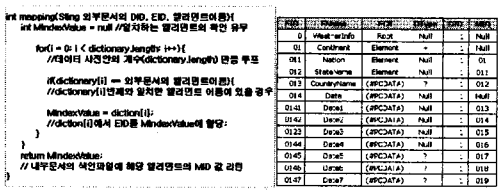
두 문서를 매핑하는 방법은 먼저 표준 DTD 문서의 색인정보 파일에서 최상위 엘리먼트의 EID를 순서로 하향식으로 외부 DTD 문서와 매핑작업을 수행한다. 이때 표준 DTD 문서의 색인정보 파일의 EID가 의미하는 EName을 데이터사전을 검색하여 의미적으로 같은 엘리먼트이름을 추출한 다음, 외부 DTD 문서의 색인정보의 EName을 검색한다. 만약 의미적으로 같은 엘리먼트 이름이 없을 경우 사용자는 시스템에서 제공하는 인터페이스를 이용하여 수동으로 매핑정보를 입력한다. 하지만 데이터 사전을 이용할 경우 의미적으로 같은 데이터가 없을 경우, 외부 DTD 문서 구조정보에 어떠한 영향도 주지 않는다. [그림8]은 이러한 두 문서를 매핑하기 위한 매핑 순서와 데이터 사전에 대한 구성요소를 표시하고 있다.



[그림8]두 문서를 매핑하기 위한 구성요소

재구성 시스템에서 제공하는 데이터 사전은 데이터베이스에 정보를 저장하지 않고 XML 파일을 데이터베이스

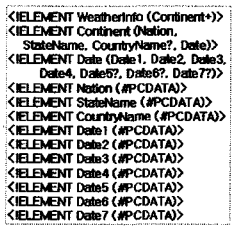
로 대체하여, 직접 XML 파일에서 정보를 검색하고 새로운 어휘를 저장한다. 데이터 사전의 구성요소를 살펴보면, DD.XML 파일은 여행정보와 관련된 항공, 호텔, 날씨에 대한 어휘정보를 저장한 파일이다. 이 파일은 XSL 문서를 이용하여 어휘내용을 HTML로 변환하고, DD.ASP 파일을 이용하여 어휘의 내용을 표시한다. 사용자가 새로운 어휘를 추가하고자 할 경우 WRITE.HTML 파일을 이용하여 새로운 어휘를 추가할 수 있으며, 추가에 대한 정보는 WRITE.ASP 파일을 이용하여 DD.XML 파일에 저장한다. 이러한 방법으로 외부 DTD 문서와 표준 DTD 문서를 매핑 하기위한 알고리즘과 표준 DTD 문서에 MID가 추가된 색인파일은 [그림9]와 같다.



[그림9]두 문서의 매핑을 위한 알고리즘과 결과

3.4. 통합된 문서의 재구성

항공, 호텔, 날씨 정보의 표준 DTD 문서를 하나의 파일로 재구성하는 부분으로서 [그림10]은 여행에 대한 정보를 제공하기 위한 3개의 관련정보를 하나의 문서로 재구성한 표준 DTD 파일을 보여주고 있다.



[그림10]하나의 문서로 재구성된 표준 DTD

3.4. 다양한 서비스를 위한 표현부분

XML 문서를 다양한 포맷으로 서비스하기 위해 CSS 파일 보다는 XML 어플리케이션인 XSL을 이용한다. 따라서 표현 부분은 XSL 표준에 기반하여 위에서 점진적인 과정을 거쳐 생성된 재구성 정보를 다양한 포맷으로 서비스를 지원한다.

4. 결론

본 논문에서는 다양한 공급자에 의해 독립적으로 운용 및 서비스되고 있는 정보를 하나의 통합된 문서로 재구성하기 위한 시스템을 설계하였다. 제안된 재구성 시스템은 서비스 제공자에게는 재배포 과정을 간소화하고 데이터 교환의 최적화는 물론 정보 교환 능력을 증가시킨다. 사용자의 측면에서는 원하는 정보를 쉽고 편리하게 제공받을 수 있는 서비스 환경을 제공한다. 그러나, DTD는 XML 문법과 같지 않아 XML 문서의 파서를 같이 사용할 수 없는 단점이 있다. 따라서 구조화되지 않거나 반구조적인 정보파일의 재구성과 XML Schema를 이용하는 문서의 재구성하는 방법에 관한 연구가 필요하다.

참고문헌

[1] http://www.w3.org/TR/REC-xml , Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 2000. 10. 6.

[2] G. Gardarin, A. Mensch, T. Tuyet Dang-Ngoc, L. Smit, "Integrating Heterogeneous Data Sources with XML and XQuery.", Proceedings of 2002 13th IEE(DEXA'02) Workshop, Page 839-846, 2002

[3] Wustner, E., Hotzel, T., Buxmann, P., "Converting business documents:a classification of problems and solutions using XML/XSLT", Advanced Issues of E-Commerce and Web-Based Information Systems(WECWIS 2002), Page 54 -61, 2002.

[4] Y.K. Lee., "Index structures for structured document.", Proceedings of the first ACM international conference on Digital libraries, Page 91-99, 1996. 4.

[5] Roy Goldman, Jason McHugh,Jennifer Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language.", Proceedings of the 2nd International Workshop on the Web and Databases(WebDB '99), 1999. 6.

[6] 황병현, 김연혜, "XML 스키마 발전 동향", 한국정보처리학회지, 제8권 제3호, Page 3-9, 2001. 5.

[7] 이경하, 이규철, "XML 프로토콜", 한국정보과학회지, 제9권 제1호, Page 31-37, 2001. 1.