

이력데이터를 이용한 XML 문서의 개선된 버전관리 방법

김성록, 박희숙, 조우현
부경대학교 컴퓨터공학과
e-mail:rok1004@mail1.pknu.ac.kr

Advanced Version Management Method for XML Documents using Historical Data

Seong-Rok Kim, Hee-Sook Park, Woo-Hyun Cho
Dept of Computer Engineering, Pukyong National University

요 약

XML은 인터넷 웹문서, EDI 등을 포함하여 다양한 분야에서 활용하는 문서이다. XML은 플랫폼에 독립적이며 문서의 정보를 손실 없이 전송 및 교환이 쉽고 웹상에서 구조화된 문서의 전송이 가능하다. 따라서 시간이 변화함에 따라 XML 문서의 변경에 따른 효율적인 버전 관리에 대한 중요성이 대두되었다. 본 논문에서는 XML문서 버전관리에 있어서 시간적인 정보를 제공하기 위해 이력 데이터베이스의 정보를 이용하고 문서의 빠른 재구성을 위해 Dewey ordering 방식을 결합한 새로운 버전번호 관리방식을 제안한다.

1. 서론

오늘날 많은 정보시스템들은 웹을 기반으로 구축되어 활용되고 있으며 다양한 전자 문서들을 이를 통해 제공하고 있다[4]. 그 대표적인 전자 문서인 XML(eXtensible Markup Language)은 W3C에 의해 표준으로 제정 및 발표된 대표적인 텍스트 포맷이다 [3]. XML은 HTML(Hypertext Markup Language)에 비해 자신의 마크업 언어를 정의할 수 있기 때문에 실제로 진정한 메타-마크업 언어라 할 수 있다. 현재 XML은 인터넷을 포함한 다양한 곳에서 데이터 교환 및 저장을 위해 실질적인 표준으로 사용되고 있다.

이처럼 XML로 저장되고 교환되는 데이터양이 증가함에 따라 XML문서에 대한 효율적 관리와 효율적 버전관리에 대한 많은 연구들이 이루어지고 있다 [2]. XML 버전관리 시스템은 지속적인 버전관리를 할 수 있는 기능이 제공하여야 한다. XML 문서에 표현된 모든 정보는 시간이 경과됨에 따라 내용의 추가, 변경, 삭제 등으로 인해 계속적으로 변화하게 된다. 이렇게 계속적으로 변하는 XML 문서의 각

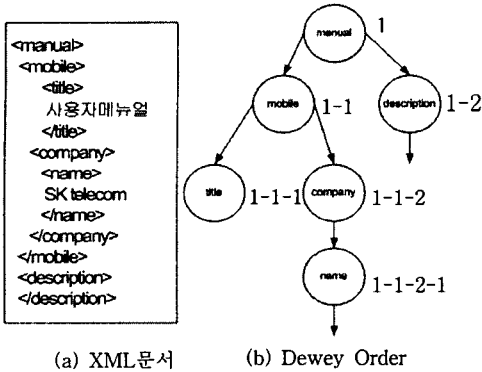
엘리먼트들에 대한 유지관리가 필요하다. 이에 따라서 본 논문에서는 버전을 부여하고 저장 관리함에 있어서 변경된 최신의 문서에 대한 정보뿐 아니라 변경 이전의 이력까지 검색이 가능하게 하는 새로운 버전관리 방법을 제안한다. 또한 제안된 방법에서는 사용자의 질의 유형에 따라 효과적으로 XML문서를 재구성하거나 특정 엘리먼트의 버전 변화 이력에 대한 정보까지도 제공해 준다.

본 논문에서는 2절에서는 관련연구를 중심으로 기술하고 3절에서는 제안하는 버전 관리 시스템에 대하여 설명을 한다. 또한 4절에서는 결론 및 향후 연구에 대해서 기술한다.

2. 관련 연구

2.1 Dewey Order 방법 및 이력데이터베이스

Dewey Order encoding 방법은 dewey 10진 분류법을 기반으로 한다. 예를 들어 XML 문서의 구성이 (그림) 1의 (a)와 같다면 이것을 Dewey Order encoding 방법을 이용해서 XML의 계층관계를 그림 1의 (b)처럼 나타낼 수 있다[5].



(그림 1) XML문서와 Dewey Order 방법 예

이력데이터베이스는 과거뿐만 아니라 미래에 관련된 데이터를 포함한다. 즉, 이력데이터베이스의 한 데이터는 이력 데이터를 포함함으로써 정의된다. 즉, 시간적인 특성들을 제공하며 시간이 변하는 데이터를 포함한다. 이력데이터베이스를 이용하는 경우 시간에 대한 질의를 할 수 있다는 특징을 갖는다. 또한, 이력데이터는 새로운 지식을 마이닝하기 위해 유용하게 사용되고 있다[6].

2.2 기존의 버전관리 시스템 소개 및 문제점

XML 문서에 대한 효율적인 관리와 검색 및 저장을 지원하는 XML 문서 버전관리 시스템은 반드시 정보 시스템 응용 분야에서 필요하다.

XML 문서를 생성하고 개발하는 동안 문서 내의 구성 요소는 변경된다. 이러한 변경 작업으로 XML 문서의 구성 요소는 변경되며 한 버전들을 다음 버전으로 변환 시키는 일련의 과정을 버전관리라 한다. 기존의 버전관리 시스템들로는 SCCS(Source Code Control System, RCS(Reverion Control System), UBCC(Usefulness Basec Copy Control), 수평 및 수직 버전 중심의 XML 버전관리 시스템들이 있다[1][2]. 이들 중에서 수직 및 수평 버전 중심의 XML 버전관리 시스템에 대해 살펴보면 다음과 같다. 수직 및 수평 버전 중심의 XML 버전관리 기법은 손실 없는 XML문서의 저장, 수정, 관리를 위한 수직과 수평 버전들을 관리한다[2]. 기존의 방법은 버전 번호 구분을 릴리스 버전번호와 변경 버전번호로 나누어 버전 번호를 유지한다. 이 방법의 장점은 XML문서의 구조를 논리적으로 관리하기 위한 버전번호 할당 방법으로 수직 버전 번호와 수평 버전번호로 나누어 표현함으로써 버전의 변경사항을 손실 없이 관리 할 수 있다는 것이다. 그러나 수직

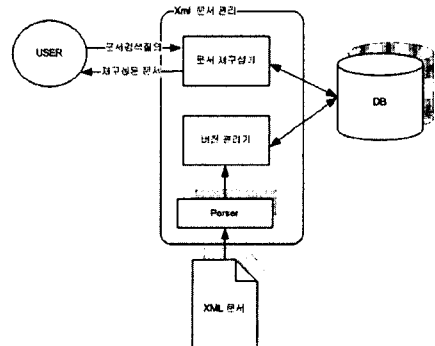
및 수평 번호 할당법만으로 XML문서의 시간적인 이력을 자세히 알 수 없고 재구성시 엘리먼트의 순서와 엘리먼트 깊이에 대한 정보를 따로 유지해야 하기 때문에 비효율적이다.

따라서 본 논문에서는 XML 문서의 버전을 지속적으로 유지하기 위해 이력데이터베이스의 이력정보와 기존의 Dewey Ordering 방법을 활용한다.

3. 제안하는 XML 문서 버전 시스템

3.1 XML 문서 버전 관리 시스템 구조

본 논문의 전체 시스템 구성도는 (그림 2)와 같다. 전체 시스템 구성도에서 버전 관리는 파서에 의해 파싱된 XML문서를 엘리먼트 단위로 분할하여 저장하는 기능과 각 엘리먼트에 버전번호를 부여하는 기능 그리고 XML문서의 변경이 발생할 때 그것을 인식하여 수정된 버전번호를 부여하고 이를 버전 테이블에 저장하는 기능을 수행한다.



(그림 2) 전체 시스템 구성도

문서 재구성기는 사용자가 요구하는 다양한 형태 질의에 대해 버전테이블의 정보를 이용하여 그 결과를 사용자에게 제공하는 역할을 수행한다. 결과로 전달되는 것들은 특정한 시점에서 재구성된 XML문서 또는 특정 엘리먼트에 대한 시간적인 버전의 변화 정보나 해당 엘리먼트의 버전번호별 버전닝 변화 과정 정보가 된다. 제안된 시스템의 데이터베이스는 (그림 3)과 같이 버전 테이블과 엘리먼트 테이블로 구성된다.

Xid	element	version	Xorder	content
-----	---------	---------	--------	---------

(a) 엘리먼트 테이블

Xid	element	version	Xorder	start_time	end_time	content	Flag
-----	---------	---------	--------	------------	----------	---------	------

(b) 버전 테이블

(그림 3) 엘리먼트 테이블과 버전 테이블

(그림 3)의 두 개의 테이블에서 Xid는 XML 문서에 부여되는 고유 식별자를 element는 XML 문서에 해당 하는 엘리먼트의 이름을 version은 버전관리기에 의해 부여되는 시간적 정보를 저장한다. Xorder는 dewey order 방법을 이용한 XML 문서의 구조에 대한 정보를 저장하며 content는 엘리먼트가 포함하는 내용이다. start_time은 각 엘리먼트의 최초 생성시간을 end_time은 각 엘리먼트의 삭제 또는 업데이트 시점을 나타낸다. 즉, end_time은 엘리먼트의 변경작업인 업데이트나 삭제, 삽입 작업이 발생할 때 새로운 버전의 엘리먼트의 생성과 연관을 가진다. 또한 Flag는 문서의 버전 변경이 일어날 때 삽입시 I, 삭제시 D, 업데이트시 U로 나타낸다.

3.2 새로운 시간정보 버전번호 부여 방법

본 논문에서는 제안하는 문서 버전할당 방법은 시간에 대한 이력정보가 포함된 버전 번호 할당 방법을 제안한다. 즉, 날짜정보와 일련의 버전번호를 조합하여 버전번호를 할당한다. 이 버전번호 할당방법은 버전 번호만을 조회하는 것으로도 XML문서내의 각 엘리먼트들의 시간적인 Versioning 변화와 버전이 일어난 횟수를 알 수 있다. 예를 들어 최초로 생성된 XML 문서에서 엘리먼트들을 데이터베이스에 저장 할 때 각 엘리먼트는 자신의 첫 번째 버전임을 나타내기 위해 버전번호로 일련번호 1을 부여받는다. 그리고 버전 번호와 함께 시간적 정보를 제공하기 위해 생성 시점의 년도 와 월을 같이 명시해 준다. 이들을 결합한 형태로는 2004년 08월 05일 최초로 생성된 엘리먼트라고 한다면 버전 번호로 0408_1로 버전 번호가 부여된다.

3.3 버전 테이블에 변경된 XML 문서저장

XML문서 (그림 1)의 (a) 예제 XML문서에 대한 엘리먼트 테이블의 내용은 (그림 4)와 같고 버전 테이블은 (그림 5)와 같다.

Xid	element	version	Xorder	content
1	manual	0407_1	1	
1	mobile	0407_1	1-1	
1	title	0407_1	1-1-1	사용자 매뉴얼
1	company	0407_1	1-1-2	
1	name	0407_1	1-1-2-1	SK telecom
1	description	0408_1	1-2	

(그림 4) 예제 XML 문서의 엘리먼트 테이블

XML문서에 대해 변경이 발생한다면 즉, 삽입, 수정, 삭제가 일어난 경우 버전 테이블에 변경된 정보

를 저장하게 되며 테이블의 내용은 (그림 6)과 같다.

Xid	element	version	Xorder	content	start_time	end_time	Flag
1	manual	0407_1	1		2004-07-09-11:11	now	
1	mobile	0407_1	1-1		2004-08-09-11:11	now	
1	title	0407_1	1-1-1	사용자 매뉴얼	2004-07-09-11:11	now	
1	company	0407_1	1-1-2		2004-07-09-11:11	now	
1	name	0407_1	1-1-2-1	SK telecom	2004-07-09-11:11	now	
1	description	0407_1	1-2		2004-07-09-11:11	now	

(그림 5) 버전 테이블

엘리먼트 하위의 menu 엘리먼트와 title 엘리먼트를 삽입하고 title 엘리먼트의 내용으로 '기본기능'이라는 내용을 삽입, company 엘리먼트의 하위 엘리먼트인 name 엘리먼트를 삭제, 그리고 company 엘리먼트의 내용을 'SK telecom'으로 내용을 수정하는 경우를 표현한 것이다.

Xid	element	version	Xorder	content	start_time	end_time	Flag
1	manual	0407_1	1		2004-07-09-11:11	now	
1	mobile	0407_1	1-1		2004-07-09-11:11	now	
1	title	0407_1	1-1-1	사용자 매뉴얼	2004-07-09-11:11	now	
1	company	0407_1	1-1-2		2004-07-09-11:11	2004-08-12-18:00	
1	name	0407_1	1-1-2-1	SK telecom	2004-07-09-11:11	2004-08-15-18:00	
1	description	0407_1	1-2		2004-07-09-11:11	now	
1	menu	0408_1	1-2-1		2004-08-01-10:00	now	I
1	title	0408-1	1-2-1-1	기본기능	2004-08-01-10:00	now	I
1	name	0408-2	1-1-2-1		2004-08-12-18:00	Expire	D
1	company	0408-2	1-1-2	SK telecom	2004-08-15-18:00	now	U

(그림 6) 변경작업 후 버전 테이블

변경 작업은 엘리먼트 단위로 버전이 일어난다. 예를 들어 2004년 8월 2일에 엘리먼트 company에 수정이 일어났다면 먼저 해당 엘리먼트의 현재 버전 번호를 가지는 레코드의 end_time필드의 값을 변경이 일어난 날짜정보로 수정한다. 다음으로 Flag 필드의 값으로 U를 version필드의 값은 엘리먼트에 수정이 일어난 날짜의 년월 정보와 이것이 해당 엘리먼트의 두 번째 버전이라는 것을 나타내기 위해 version필드의 값으로 시간정보의 0408과 일련번호 2를 결합하여 0408_2를 버전 번호를 부여하고 이것을 버전 테이블에 저장한다. 이것은 버전 번호에 시간정보를 결합함으로써 버전이 변화된 시간적인 진

행 과정을 쉽게 파악할 수 있으며 또한 버전의 일어난 횟수도 쉽게 알 수 있는 장점을 가진다. 새로운 엘리먼트의 삽입시에는 flag 필드 값으로 I를 저장하고 현재의 시간정보와 일련번호 1을 결합한 값을 버전번호로 부여하고 start_time 필드 값으로는 생성시간을 end_time 필드 값으로는 now를 준다. 특정 엘리먼트에 삭제가 발생할 경우 기존의 버전번호에서 시간정보와 버전번호를 증가 시키고 end-time 필드에는 삭제가 완전히 되어진 표시로 'Expire'를 표시 한다.

3.4 사용자 중심의 다양한 질의 수행

본 논문에서 제안한 방법은 사용자가 원하는 다양한 유형의 질의 수행이 가능한 효과적인 구조를 가진다. 특히 특정 엘리먼트의 전체 버전 변경횟수에 대한 정보 검색을 질의하는 경우 또는 특정 엘리먼트의 시간적인 버전 진화과정에 대한 정보 또는 사용자가 원하는 특정 시점에 해당하는 문서의 재구성을 원할 때 효율적인 XML 문서의 재구성을 지원한다. 질의 유형별 예는 다음과 같다.

질의유형 1 : 현재시점을 기준으로 변경된 XML문서의 재구성을 원하는 경우

```
<manual>
<mobile>
<title>사용자 매뉴얼</title>
<company>SK telecom</company>
</mobile>
<description>
<menu>
<title> 기본 기능 </title>
</menu>
```

(그림 6) 현재시점의 XML문서 재구성

질의유형 2 : 과거의 특정 날짜(2004-08-13)까지 변경된 문서의 재구성을 원하는 경우

```
<manual>
<mobile>
<title>사용자 매뉴얼</title>
<company></company>
</mobile>
<description>
<menu>
<title> 기본 기능 </title>
</menu>
```

(그림 7) 과거 특정시점 기준의 XML문서 재구성 위의 (그림 7)에서 과거의 어느 시점을 기준으로 그 문서의 구성을 원할 때 예를 들면, start_time이 2004-08-13인 것을 기준으로 적거나 같은 모든 레코드들을 버전 테이블로부터 검색하여 문서를 재구성할 때 Xid 필드와 Xorder 필드 기준으로 정렬 한 후 같은 Xorder 필드의 값을 가질 경우 version 번

호를 비교해서 큰 값을 가지는 엘리먼트를 Xid별로 문서를 재구성 할 때 포함 시킨다. (그림 7)에서 보는 것과 같이 company 엘리먼트의 텍스트는 재구성을 원하는 날짜 이후에 생성 된 것이므로 문서에 포함되지 않음을 확인할 수 있다.

질의 유형 3 : 특정 엘리먼트 버전닝 과정

(그림 6)에서 company 엘리먼트에 대한 날짜별, 버전 번호별 질의 처리 결과는 (그림 8)과 같이 표현한다.

```
날짜별 company의 버전과정
2004-07-09-11:11 -> 2004-08-15-18:00
버전번호별 company의 버전과정
0407_1 -> 0408_2
```

(그림 8) 특정 엘리먼트 버전과정

4. 결론 및 향후방향

본 논문에서는 이력데이터베이스의 이력정보를 활용한 새로운 XML 문서의 버전 관리방법을 제안하였다. 또한 기존의 방법과 비교해서 사용자들이 시간의 흐름에 따라 이력 정보 및 질의를 다양하게 할 수 있음을 보였다. 앞으로의 연구 과제로는 제안된 버전관리 시스템 구현 및 시스템에 대한 성능 평가 수행이 요구 된다.

참고문헌

- [1] Shu-Yao Chien, Vassilis J. Tsotras, Carlo zaniolo, Donghui Zhang, "Storing and Querying Multiversion XML Documents using durable Node Numbers", In Proc. 2002 IEEE
- [2] 배양석, 손충범, 유재수, "XML문서 관리를 효율적인 버전관리 방법", 2002.
- [3] 이홍래, 이형동, 유상원, 김형주, "XML 문서상에서 키워드 검색을 위한 색인 분할 기법", 한국정보과학회, 2004.5
- [4] 김성준, 김동아, 이석균, "X-treeDiff를 이용한 XML 문서의 버전 관리 시스템 프로토타입 개발", 한국정보처리학회 추계학술 발표대회 논문집 제 10 권 제2호, 2003.11
- [5] Igor Tatarinov, Stratis D. Viglas, Kevin Beyer, Jayavel Shanmugasundaram, Eugene Shekita, Chun Zhang, "Storing and Querying Ordered XML Using a Relational Database System", ACM SIGMOD'2002
- [6] C.J Date, "An introduction to Database Systems", Pearson Addison Wesley, 2004