

XML 실체 뷰 관리 프레임워크

임재국^{○*} 문찬호⁺ 김영성⁺ 강현철⁺ 서상구⁺⁺

[○]중앙대학교컴퓨터공학과

⁺⁺광운대학교경영정보학과

{jklim, moonch, yskim, hckang}@rose.cse.cau.ac.kr, skseo@daisy.gwu.ac.kr

A Framework of XML Materialized View Maintenance

JaeGuk Lim^{○*} ChanHo Moon⁺ YoungSung Kim⁺ Hyunchul Kang⁺ Sang-Koo Seo⁺⁺

⁺Dept. of Computer Science and Engineering, Chung-Ang University

⁺⁺Dept. of MIS, Kwangwoon University

요 약

W3C(World Wide Web Consortium)가 XML(eXtensible Markup Language)을 차세대 웹 문서 표준으로 제안하면서 많은 응용에서 XML을 사용하려는 다양한 연구가 진행되고 있다. 뷰는 이질적인(heterogeneous) 데이터의 통합 및 여과(filtering) 기능을 통해서 사용자가 요구하는 데이터를 제공한다. 뷰의 구현 기법 중에서 실체 뷰(materialized view)는 뷰의 내용을 직접 저장하는 기법으로서, 사용자의 질의에 따른 응답시간을 제공할 수 있다. 그러나 실체 뷰는 하부 데이터베이스가 변경되었을 경우에 일관성을 유지해야 하는 오버헤드가 있다. XML은 문서의 구조정보를 나타낼 수 있으므로 기존의 관계 데이터베이스와 객체지향 데이터베이스 등에서 연구되었던 실체 뷰 관리 기법과는 다른 XML 실체 뷰 관리 기법이 요구된다. 본 논문에서는 XML 문서를 대상으로 XML 실체 뷰를 지원하고, 데이터베이스에 저장된 XML 문서가 수정되었을 경우에 XML 실체 뷰에 대해 점진적 갱신(incremental refresh)을 지원하는 XML 실체 뷰 관리 프레임워크(framework)를 제안한다.

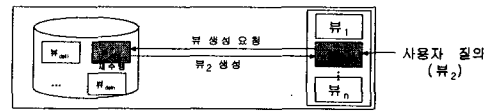
1. 서론

W3C(World Wide Web Consortium)에 의해 XML(eXtensible Markup Language)이 차세대 웹 문서 표준으로 제안되면서, 웹 문서, EDI(Electronic Data Interchange), E-Commerce 등과 같은 다양한 응용에서 XML 문서를 활용하려는 노력이 진행 중에 있다. 이에 XML 문서의 구조정보 표현 방법 및 저장 시스템에 대한 연구가 이루어지고 있으며, XQL(XML Query Language) 등과 같은 XML 표준 질의어에 대한 연구도 진행되고 있다.

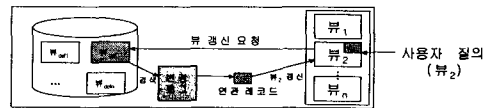
인터넷의 확산으로 현재 웹에서는 다양한 데이터 저장소(repository)에 많은 양의 데이터들이 산재되어 있다. 뷰는 이질적인 데이터의 통합과 데이터 여과(filtering) 기능을 통해서 사용자가 요구하는 데이터를 제공한다. 뷰에 관해서는 관계 데이터베이스, 객체지향 데이터베이스 시스템 등에서 다양한 연구가 진행되었다.

뷰를 구현하는 기법에는 질의 변경(query modification)과 실체 뷰(materialized view)가 있다. 질의 변경은 뷰의 정의만을 시스템 목록에 저장해 놓고, 뷰에 대한 요청이 있을 때마다 그림(1)의 (a)와 같이 뷰 전체를 재생성하는 기법으로 뷰를 재생성하는 데 일반적으로 많은 시간이 걸릴 수 있다. 질의 변경과는 달리 실체 뷰 기법은 뷰의 내용을 실제로 저장하는 기법으로서, 사용자의 질의에 따른 응답시간을 제공할 수 있다. 그러나 실체 뷰는 하부 데이터가 변경되었을 경우에 일관성을 유지해야 하는 오버헤드가 있는데, 뷰의 제공이 요청되었을 때(그림 1)의 (b)와 같이 뷰의 일부분만을

갱신하는 점진적 갱신(incremental refresh) 기법을 사용함으로써 뷰 갱신 오버헤드를 줄일 수 있다.



(a) 질의 변경에서 뷰 제공



(b) 실체 뷰에서 점진적 뷰 갱신

(그림 1) 질의 변경과 실체 뷰

본 논문에서는 XML 문서를 대상으로 실체 뷰를 지원하고 XML 문서가 수정(update)되었을 경우에 XML 실체 뷰에 대해 점진적 갱신을 지원하는 XML 실체 뷰 관리 프레임워크(framework)를 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 XML과 뷰에 대한 관련연구를 기술한다. 3절에서는 본 논문에서 제안한 XML 실체 뷰 관리 프레임워크에 대해 기술한다. 마지막으로 4절에서 결론을 맺는다.

2. 관련연구

실체 뷰에서는 하부 데이터베이스에서 데이터 수정이 발

* 본 논문은 정보통신부의 대학 정보통신 연구센터 육성·지원 사업에 의한 것임.

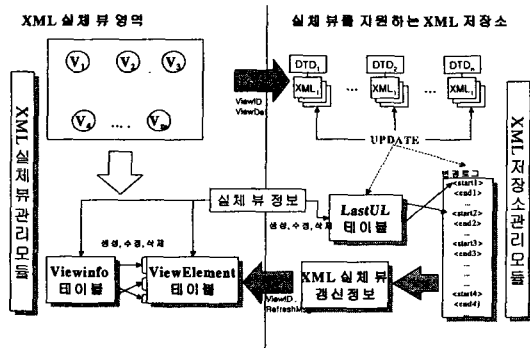
생했을 경우에 뷰와 데이터베이스 간의 일관성 유지가 중요한 문제이다. [1][2]에서는 데이터베이스에서 수정된 데이터 중에서 실제 뷰에 영향을 미치는 레코드(relevant record)를 찾아서 실제 뷰의 일부분만을 갱신하는 지연된 점진적 갱신(deferred incremental refresh) 기법을 제안하였다. 이 기법은 실제 뷰 갱신 오버헤드를 줄일 수 있다. 최근에 [3]에서는 반구조적(semistructured) 데이터 관리 시스템인 Lore 환경에서 실제 뷰 관리 및 점진적 갱신 방법을 제안하였다.

XML은 문서의 구조정보를 나타낼 수 있으므로 XML 실제 뷰를 관리하기 위해서 기존에 연구되었던 실제 뷰 기법과는 다른 기법이 요구된다. 최근에 XML 뷰에 대한 연구가 진행되고 있지만[4][5], XML 실제 뷰에 관한 연구는 미비한 상태이다.

3. XML 실제 뷰 관리 프레임워크

본 논문에서 제안하는 XML 실제 뷰 지원 프레임워크는 (그림 2)와 같다. XML 실제 뷰 영역에는 각 XML 실제 뷰에 대한 정보, EID(element ID) 사상(mapping)정보와 실제 뷰의 내용이 저장된다. 실제 뷰 지원 XML 저장소에는 다수의 DTD와 XML 문서, XML 실제 뷰의 점진적 갱신을 지원하기 위해서 XML 문서의 수정 내용에 대한 변경로그(update log), 각 XML 실제 뷰와 변경로그의 연결정보가 저장된다.

XML 실제 뷰 관리 모듈은 뷰에 대한 갱신 요청이 발생하면 XML 저장소 관리 모듈에게 XML 실제 뷰 갱신정보를 요청한다. 갱신정보 요청을 받은 XML 저장소 관리 모듈은 변경로그를 검색해서 XML 실제 뷰 갱신정보를 생성하고, 이것을 XML 실제 뷰 관리 모듈에게 전달한다. XML 실제 뷰 갱신정보가 전달되면 XML 실제 뷰 관리 모듈은 이 갱신정보를 이용해서 해당 XML 실제 뷰에 대한 점진적 갱신을 수행한다.



(그림 2) XML 실제 뷰 관리 프레임워크

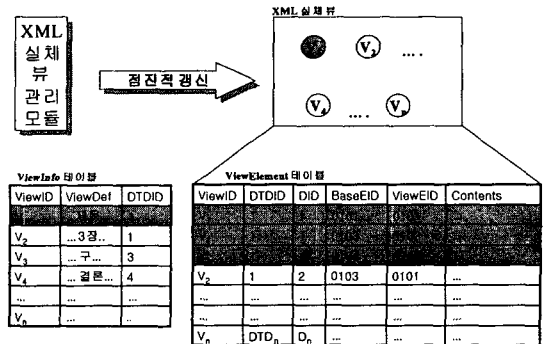
본 절에서는 본 논문에서 제안하는 XML 실제 뷰 관리 프레임워크의 몇 가지 가정을 기술하고 XML 실제 뷰 영역과 실제 뷰를 지원하는 XML 저장소에 대해 기술한다.

3.1 가정

본 논문에서 제안하는 XML 실제 뷰 관리 프레임워크에서는 다음을 가정하였다.

- ① 각 XML 실제 뷰는 단일 DTD를 대상으로 생성된다. 즉, XML 저장소에 다수의 DTD가 저장되어 있고 각 DTD와 연관된 다수의 XML 문서들이 저장되어 있을 때, XML 실제 뷰는 하나의 DTD와 연관되어 저장된 XML 문서들을 대상으로 뷰를 생성한다.
- ② XML 문서의 수정은 엘리먼트 단위로 일어난다.

3.2 XML 실제 뷰 영역



(그림 3) XML 실제 뷰 영역 구조

XML 실제 뷰 영역에는 (그림 3)과 같이 다수의 뷰가 존재하며, 각 뷰는 ViewInfo 테이블과 ViewElement 테이블에 의해 관리된다. XML 실제 뷰 관리 모듈은 뷰에 대한 갱신 요청이 발생했을 때, 갱신정보를 요청하고 해당 뷰의 점진적 갱신을 수행한다.

ViewInfo 테이블에는 각 XML 실제 뷰에 대해 1개의 레코드가 존재한다. ViewInfo 테이블에 저장되는 정보는 XML 실제 뷰를 구별하는 ViewID, 뷰의 생성 조건을 정의한 뷰 정의(view definition), 뷰가 참조하는 DTD의 ID(DTID) 등이다.

ViewElement 테이블은 ViewInfo 테이블과 ViewID와 DTID로 연결된다. DID는 뷰에 저장된 엘리먼트의 XML 문서 번호를 나타낸다. XML 저장소에 저장되었던 엘리먼트의 EID를 뷰에 저장된 엘리먼트의 EID로 사상시키기 위해서 BaseEID와 ViewEID를 사용한다¹⁾. 이것은 사용자가 XML 저장소를 대상으로 질의를 제기했을 경우, 뷰에 저장된 엘리먼트로 EID를 사상시키기 위해서 필요하다. Contents에는 실제 XML 문서의 데이터가 저장되는데, 시스템이 중앙 집중 환경이라면 데이터 중복을 방지하기 위해서 (DTDID, DID, EID)가 저장되고, 분산 환경이라면 실제 값이 저장된다.

3.3 XML 실제 뷰를 지원하는 XML 저장소 구조

XML 저장소에는 (그림 2)과 같이 다수의 DTD와 XML 문서들이 저장된다. XML 저장소 관리 모듈은 XML 저장소에서 XML 문서가 수정되었을 때, XML 실제 뷰의 점진적 갱신을 지원하기 위해서 수정 내용에 대한 변경로그를 기록한다. 또한 XML 실제 뷰 갱신정보의 요청이 있을 때, 갱신정보를 생성하여 XML 실제 뷰 관리 모듈로 전달한다.

```
<Start UpdateLog, DID, OpType, ObjType>
<BaseEID, (REID [{"Value"(DTID, DID, EID)}, REID] )>
<End UpdateLog>
```

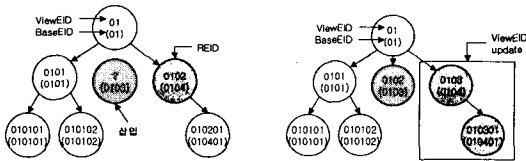
(그림 4) 변경로그 레코드 자료구조

변경로그 레코드의 자료구조는 (그림 4)와 같다. 변경로그 레코드는 <Start UpdateLog>와 <End UpdateLog>의 쌍으로 구성된다. 이것은 XML 문서에 대한 일련의 수정을 하나의 트랜잭션으로 간주하기 위한 것이다. DID는 실제 수정된 XML 문서의 ID를 나타낸다. OpType은 XML 문서에 대한 수정의 종류를 나타낸다. OpType이 '1'이면 엘리먼트 내용 수정,

1) 기존 논문들에서 EID 할당에 대한 많은 연구가 진행되었다. 본 논문에서는 [6]에서 제안한 방법을 이용해서 EID를 할당한다.

'2' 이면 새로운 엘리먼트 삽입, '3' 이면 엘리먼트 삭제 를 의미한다. ObjType 은 XML 문서에서 수정되는 단위를 나타 낸다. ObjType 이 '1' 이면 문서 단위, '2' 이면 엘리먼트 단 위, '3' 이면 속성 단위로 수정되는 것을 의미한다.

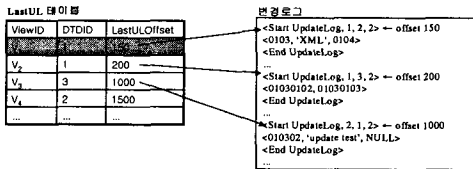
변경로그 레코드의 내부에 저장되는 <BaseEID, (REID | ('Value' | (DTDID, DID, EID))[, REID])> 는 수정된 엘리먼트 에 대한 정보이다. BaseEID는 수정된 엘리먼트의 EID를 나 타낸다. XML 문서에 엘리먼트의 삽입 혹은 삭제가 발생하면, 문서의 구조정보가 변경되고 이것은 XML 실제 뷰에 반영되어 야 한다. 이를 위해 변경로그에서는 REID(Relative EID) 를 사용한다.



(그림 5) XML 문서 수정 시에 ViewEID의 변화

(그림 5)와 같이 BaseEID가 '0101' 과 '0104' 인 두 엘 리먼트 사이에 EID가 '0103' 인 엘리먼트가 삽입되면, 삽입 된 엘리먼트의 오른쪽 형제 엘리먼트들과 그 자손 엘리먼트 들의 ViewEID는 영향을 받는다. 이때, REID를 이용해서 추가 된 엘리먼트의 오른쪽 첫번째 형제 엘리먼트의 EID를 XML 실 체 뷰 관리 모듈에게 전달하면, XML 실제 뷰 관리 모듈은 구조변경이 발생한 모든 ViewEID를 수정할 수 있다. 엘리먼트 의 내용 변경일 경우에는 REID는 NULL로 설정된다.

('Value' | (DTDID, DID, EID)) 는 수정된 엘리먼트의 값 을 나타낸다. 분산 환경에서는 실제 값을 저장하기 위해 Value를 사용하고, 중앙집중 환경에서는 (DTDID, DID, EID) 를 사용한다.



(그림 6) LastUL 테이블 구조 및 변경로그와의 관계

(그림 6)은 XML 실제 뷰와 변경로그 간의 연결정보를 저 장하고 있는 LastUL 테이블 구조와 변경로그와의 관계를 나 타낸 것이다. LastUL 테이블에는 뷰를 구별하기 위해 서 ViewID가 저장되고, 뷰가 참조하는 DTDID가 저장된다. 또한 뷰와 변경로그를 연결하기 위해서 LastULOffset을 저장한다. LastULOffset은 해당 실제 뷰의 XML 실제 뷰 갱신정보를 생 성하기 위해서 가장 마지막으로 읽은 변경로그 레코드의 다 음 오프셋(offset)을 가리킨다. 이를 통해 다음 갱신이 요구될 때, LastULOffset이 가리키는 곳에 기록된 변경로그 레코드 부터 마지막 변경로그 레코드까지 검색함으로써, 변경로그의 검색 양을 줄일 수 있다. 실제 뷰가 생성될 때 LastULOffset 은 그 시점에서 변경로그의 끝 오프셋으로 설정된다. XML 실제 뷰 갱신정보를 생성하기 위해 변경로그 검색을 마친 후에도 LastULOffset은 변경로그의 끝 오프셋으로 설정된다.

XML 실제 뷰의 갱신이 요청됐을 때, XML 저장소 관리 모듈 은 (그림 7)과 같은 구조의 XML 실제 뷰 갱신정보를 생성한 다. (그림 7)에 나타난 XML 실제 뷰 갱신정보는 XML 실제 뷰 V₁에게 전달될 XML 실제 뷰 갱신정보를 예를 들어 나타낸 것 이다. 이 예에서 첫번째 갱신정보 레코드는 XML 저장소에 저

장된 문서 중에서 문서 번호 '1' 에 EID '0103' 인 엘리먼트 가 삽입되었고, 이것은 ViewElement 테이블에 저장된 엘리 멘트 중에서 ViewID가 V₁, DTDID, DID가 '1' 이고 BaseEID가 '0104' 인 엘리먼트부터 EID에 영향을 받는다는 것을 나타 낸다.

ViewID	OpType	ObjType	DTDID	DID	EID	REID	Contents
V ₁	3	2	1	1	010302	010303	NULL
V ₁	1	2	1	1	010303	NULL	update test
V ₁	1	2	1	1	010304	010405	NULL

(그림 7) XML 실제 뷰 갱신정보 생성 예

3.4 XML 실제 뷰의 점진적 갱신 절차

XML 실제 뷰가 요구될 때, 실제 뷰 갱신 절차를 요약하면 다음과 같다

- ① XML 실제 뷰 관리 모듈은 XML 실제 뷰 갱신정보를 요청하기 위해서 ViewInfo 테이블에 저장된 해당 뷰의 ViewID와 ViewDef를 XML 저장소 관리 모듈로 전송한다.
- ② 요청을 받은 XML 저장소 관리 모듈은 LastUL 테이블에서 해당 뷰의 LastULOffset이 가리키는 변경로그 레코드부터 변경로그의 끝까지 검색해서 XML 실제 뷰 갱신정보를 생성한다. 이때, 필요한 갱신정보만을 추출하기 위해서 XML 실제 뷰 관리 모듈로부터 전송받은 ViewDef를 이용하여 변경로그와 해당 뷰의 연관성 (relevance)을 검사한다.
- ③ 갱신정보 생성이 완료되면 ViewID와 함께 갱신정보를 XML 실제 뷰 관리 모듈에게 전송한다.
- ④ XML 실제 뷰 관리 모듈은 전송받은 XML 실제 뷰 갱신 정보를 이용해서 ViewID로 지정된 실제 뷰에 대해 점진적 갱신을 수행한다.

4. 결론

XML은 문서의 구조정보를 나타낼 수 있으므로 기존의 관계 데이터베이스와 객체지향 데이터베이스 등에서 연구되었던 실제 뷰 관리 기법과는 다른 XML 실제 뷰 관리 기법이 요구 된다. 본 논문에서는 XML 문서 변경 시에, 구조변경 및 내용 변경 사항들을 점진적 갱신을 통해 효율적으로 처리할 수 있는 XML 실제 뷰 관리 프레임워크를 제안하였다.

5. 참고문헌

- [1] B. Lindsay et al., "A Snapshot Differential Refresh Algorithm," SIGMOD Record, Vol. 15, No. 3, 1986, pp. 53-60.
- [2] J. Blakeley et al., "Efficiently Updating Materialized Views," SIGMOD Record, Vol. 15, No. 3, 1986, pp. 61-71.
- [3] S. Abiteboul et al., "Incremental Maintenance for Materialized Views over Semistructured Data," Proc. Int'l Conf. on VLDB, 1998, pp.38-49.
- [4] S. Abiteboul, "On Views and XML," Proc. ACM Symp. On Principles of Database System, 1999, pp. 1-9.
- [5] Y. Papakonstantinou and V. Vianu, "DTD Inference for Views of XML Data," Proc. of 19th ACM SIGACT-SIGMOD-SIGART Symp. on PODS, 2000.
- [6] 연제원 외 3명, "XML 문서 구조 검색을 위한 저장 시스템 설계," 한국정보과학회, '99년 봄 학술발표논문집, 26권, 1호, 1999, pp.3-5.