RDB 에서의 검색 효율을 위한 XML 문서 저장 모델

권 훈* • 강 인 석* • 김 정 희* • 곽 호 영**

XML Document Storage Model for Searching Efficiency in Relational Database

Hoon Kwon*, In-seok Kang*, Jeong-Hee Kim* and Ho-Young Kwak**

ABSTRACT

The purpose of the paper is design and implementation of Internet-Based Real-Time Admission XML instances for purpose of information exchange are normally stored in the legacy relational database. therefore, integrations with relational database are required for effective XML applications. to support these requirements, virtual decomposition storage or decomposition storage methods which save separates structures of instances to relational database have researched. however, these storage methods contain different information of instance structure and layers which has caused difficulties to process query during search operation as well as increased overheads due to duplicate savings for separate storages. therefore, in this research, additional field of "Eltype" has introduced to previous database schema structure to integrate instance and instance structure, provide consistent information of layers and propose storage structure to map each field to schema field of relational database. as results, XML instance and structures can be stored together to minimize overheads and required disk space. also, synchronized storage layer structure provides easier processing of search query.

Key Words : XML, RDB, storage model

Ⅰ. 서 론

인터넷의 사용과 정보의 양의 급증에 따른 인터넷 기반의 정보들을 보다 효율적으로 이용하기 위한 연구가 활발히 이루어지고 있다. 이에 웹상의 거의 모든 정보는 HTML을 기반으로 이루어져, 문서의 구조보다는 표현에 중점을 두고 있다.

따라서, 특정 응용 분야로의 정보를 활용하는 것에는 기능이 부족하다는 단점이 있다[1,2]. 이에, W3C에서는 차세대 웹 문서의 표준으로 XML(eXtensible Markup Language)이라는 전자문서 메타언어를 1996년에 제안, 현재까지 그 기능을 계속 확장해 오고 있다[2,3].

특히 전자상거래 분야에서 XML을 이용하여 이루어지는 문서의 교환이 많아지고 있는 실정이다. 또한, 웹에서의 정보교환을 위한 XML 메

* 제주대학교 대학원
Graduate School, Cheju Nat.l Univ.
** 제주대학교 통신·컴퓨터공학부, 첨단기술연구소
Faculty of Telecommunication & Computer Eng., Res. Insti. Advanced Tech., Cheju Nat.l Univ.

시지의 소스 데이터는 Legacy 데이터베이스에 저장되어 있기 때문에, 이에 따라 XML 응용과 데이터베이스 시스템과의 원활한 연동이 요구되어진다. eXoelon과 tamino같은 XML 전용 DBMS는 XML 문서를 자동으로 저장하고 검색을 할 수 있도록 되어 있다. 그러나, 기존의 자료를 활용해야 하는 상황에서 XML문서를 기존의 데이터베이스에 저장하고 검색할 수 있도록 해야 하는 필요성이 요구되고 있다. 이러한 필요성에도 불구하고 기존에 사용하고 있는 RDBMS에서는 효율적인 저장 및 검색이 불가능하다. 이것은 XML구조와 RDB구조가 서로 상이하기 때문이다. 따라서, XML 문서를 RDB 기반에서 저장, 검색할 수 있는 구조로 변경해 주는 모델링이 필요하다[4-6].

이에 본 연구는 RDB 기반의 검색 효율성을 고려한 XML 문서의 저장구조에 따른 모델을 제안한다. 제안 모델은 문서구조의 저장과 검색을 위해 하나의 테이블에 통합하여 구조를 저장하였고, 테이블 구조는 XML 스키마 종속적인 구조를 갖는다.

본 연구의 구성은 먼저 2장에서 관련연구로 기존 XML 문서에 저장에 따르는 저장 모델 구조를 살펴보고, 3장에서는 제안 모델에 대한 기본 저장구조를 제시한다. 또한, 제시된 저장구조를 적용하여 XML문서를 저장하여 본다. 4장에서는 기존의 XML 문서 저장 모델 구조와 제안 모델 구조를 비교 분석하며, 끝으로 5장에서는 결론 및 향후과제를 기술한다.

II. 관련 연구

웹상에서의 정보들을 활용하기 위한 방법으로 사용되고 있는 XML 문서가 점차 확산됨에 따라 보다 효율적으로 XML 문서가 가지고 있는 정보를 저장하는 방법에 많은 연구들이 진행되고 있다[7,8]. 기존 XML 문서 저장 모델링은 크게 다음과 같은 방식에 의해 RDB에 저장하며, 연구되어지고 있다.

2.1. 분할 저장 모델링

분할(decomposition) 저장 모델링은 XML문서의 트리구조가 각 객체의 실제 노드내용을 가지고 저장되며, 이는 XML 문서를 엘리먼트 단위로 쪼개어 저장하고, 검색시 구조 정보를 참조하여 해당 엘리먼트나 하위 엘리먼트의 조합을 생성하여 처리하는 모델링 기법이다. 분할되어진 엘리먼트에 관계된 엘리먼트의 수정으로 문서의 편집 및 관리가 쉽지만, 문서의 내용을 검색하여 추출할때는 각 엘리먼트의 내용을 조합하여 결과를 구성해야 함으로 검색과정이 복잡하며 시간이 오래걸린다는 단점이 있다[6,7].

2.2. 가상 분할 저장 모델링

트리상의 각 노드는 실제 문서의 논리적인 구조만을 기술하고 내용부분을 다른 영역에서 관리하여 각각의 노드가 이 영역에 대한 위치정보와 길이를 가지고 내용을 표시하는 모델링 기법이 가상 분할 저장 모델링 기법이다. 문서를 한꺼번에 저장하기 때문에 분할저장에서의 통합이 불필요하고, 위치에 따르는 구조적 검색이 가능하여 검색효율이 우수하다는 장점이 있지만, 엘리먼트의 추가, 삭제시에는 위치정보의 갱신이 불가피 하다는 단점이 있다[6,7].

III. 제안 XML 문서 저장 모델링

Fig. 1은 본 연구에서 제안하는 저장 구조이며, 각각의 필드는 RDB의 데이터베이스 스키마를 따르는 필드가 된다.

3.1. 제안 구조 필드 정보

Fig. 1에서 보여지는 것과 같이 제안 저장 구조에는 많은 구조필드로 구성되어 있다. 이 필드들의 내용은 Table 1과 같다.

Table 1의 필드들 중 LevelInfo 필드는 3부분으로 나뉘어, 실제 Element에 대한 위치정보를 나타내게 된다. 이 정보는 자신의 상위를 나타내는 Root, 왼쪽 형제를 나타내는 Left, 오른쪽 형제를 나타내는 Right로 세분화 되어 있다. Eltype 필드는 Element 정의 타입형식으로 스

SICHK	ID	SID	LevelInfo			Element	Attribute	Entity	Eltype	Rep	NameSpace		Ref
			Root	Left	Right						NsName	URI	

Fig. 1 Proposed structure

키마에서 주어지는 형식을 CLOB(Charactor Large OBject)기법을 이용하여 축약시켜 가상 분할 저장하였다. 마지막으로 Ref 필드는 문서 내의 ELement 정의형식에 따른 참조를 나타내는 값으로 구성된다.

Table 1. Field information

SICHK	Identify instance and schema	S, I
ID	Identify element	-
SID	Identify number, instance and schema	-
LevelInfo	Element level information	-
Element	Element name	-
Attribute	Attribute value	-
Entity	Contents value	-
Eltype	Element definition type	CS,CA
Rep	Element repeat number	-
NameSpace	NameSpace information	-
Ref	Reference value in in document	-

3.2. 적용 예시문서

제안된 구조에 모델링을 하기 위해 Table 2와 같은 예시문서를 사용하였다.

3.3. 제안 모델링 결과

Table 2에 제시된 예제 문서를 가지고 실제 제안 구조에 저장을 보면 Table 3와 같이 RDB 스키마에 모델링 되어진다.

Table 2. Examples of XML Document

	Example 1
XML Schema	<?xml version="1.0" ?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:element name="Order"> <xs:complexType> <xs:sequence> <xs:element name="Name" type="xs:string"/> <xs:element name="Su" type="xs:string"/> </xs:sequence> </xs:compleType> </xs:element> </xs:schema>
Ins-tance	<?xml version="1.0" ?> <Order> <Name>DB book</Name> <Su>3</Su> </Order>
	Example 2
XML Schema	<?xml version="1.0" ?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:element name="Customer"> <xs:complexType> <xs:sequence> <xs:element name="FirstName" type="xs:string" /> <xs:element name="Middle" type="xs:string" /> <xs:element name="LastName" type="xs:string" /> </xs:sequence> <xs:attribute name="customerID" type="integer" /> </xs:complexType> </xs:element> </xs:schema>
Ins-tance	<?xml version="1.0" ?> <Customer customerID="24332"> <FirstName>Ray</FirstName> <MiddleInitial>G</MiddleInitial> <LastName>Bay</LastName> </Customer>

IV. 비교 분석

4.1. 기존 모델링 분석

XML 인스턴스와 스키마를 서로 분리하여 정적 테이블에 저장했을때, 많은 데이터에 따르는

중복필드로 인하여 오버헤드가 높아진다.

또한, 스키마를 저장시 엘리먼트 단위의 ID를 부여하고 있기 때문에, 엘리먼트 문서형식 정의를 위한 엘리먼트 역시 각각의 ID로 부여하고 있다. 이는 결과적으로 XML 인스턴스와 엘리먼트간의 계층정보 불일치를 나타내게 되어, 검색질의시 처리를 어렵게 만들고 있다.

4.2. 제안 모델링 분석

Table 1의 필드내용을 바탕으로 Table 2의 예제문서를 적용시켜 나타낸 모델링 결과인 Table 3을 살펴보면 제안 모델링 구조의 특징을 확인할 수 있다. 첫째, LevelInfo 를 보면, 각각의 값은 각 자신의 엘리먼트로부터 보여지는 위치를 기억하고 있게된다. 여기서, 주목해야 할 것은 XML 스키마상의 LevelInfo와 인스턴스간의 LevelInfo값이 서로 동일함을 알 수 있다.

이는 스키마의 엘리먼트 정의형식을 Eltype을 통해 CLOB화 시켜 저장하였기 때문에 가능하게 되었다. 따라서, LevelInfo와 각각의 ID값만 있다면 정확히 해당하는 엘리먼트의 컨텐츠를 검색할 수 있게 된다.

둘째, 스키마에 대한 저장시, 각 엘리먼트에 해당하는 정의구조 형식을 ID에 부여하여 저장

Table 3. Modeling result

S	0/1	0/0/1	xml	version="1.0"					
S	0/1	0/1/0	schema					xs / http://...	
S	1/1	0/0/0	element	name="Order"		CS	*	xs / http://...	
S	2/1	1/0/3	element	name="Name" type="xs:string"				xs / http://...	
S	3/1	1/2/0	element	name="Su" type="xs:string"				xs / http://...	
				... ...					
I	0/1	0/0/0	xml	version="1.0"					
I	1/1	0/0/0	Order				1		
I	2/1	1/0/3	Name		DB book				
I	3/1	1/2/0	Su		3				
				... ...					
S	0/2	0/0/1	xml	version="1.0"					
S	0/2	0/1/0	schema					xs / http://...	
S	1/2	0/0/0	element	name="Customer"		CS	*	xs / http://...	
S	2/2	1/0/3	element	name="FirstName" type="xs:string"				xs / http://...	
S	3/2	1/2/4	element	name="Middle" type="xs:string"				xs / http://...	
				... ...					
S	1A/2	0/0/0	attribute	name="customerID" type="xs:integer"				xs / http://...	
I	0/2	0/0/0	xml	version="1.0"					
I	1/2	0/0/0	Customer	CustomerID="24332"			1		
I	2/2	1/0/3	FirstName		Ray				
I	3/2	1/2/4	Middle		G				
				... ...					

하는 기존 방식과는 달리, CLOB를 통한 Eltype 필드에 따로 저장함으로써, 중복되는 정의구조 형식에 따른 오버헤드부분을 낮추게 되었다. 기존 모델링 방법과 제안 모델링 방법에 따르는 결과를 Table 4에 제시하고 있다.

Table 4. Analyzing legacy modeling and proposed modeling

Stored content	structure or content	structure and content
Method of storage	structure and instance stored separately	structure and instance stored integration
Level structure	Unidentical between structure and instance	Identical between structure and instance
Overhead of structure information	High overhead of structure information	Low overhead due to Eltype fields

V . 결 론

XML 문서를 관계형 데이터베이스에 저장하기 위한 기존 모델링 방법은 XML 스키마와 인스턴스를 분리 저장하였으며, 또한 데이터베이스 스키마는 가상 또는 분할방식의 기반위에서 여러 방식으로 연구되어왔다.

이러한 모델링 방법은 XML 스키마와 XML 인스턴스간의 계층정보 불일치로 검색시 질의처리를 어렵게 하였다.

또한, 엘리먼트를 기준으로 ID를 부여하여 데이터베이스 스키마에 저장함으로써, 엘리먼트 정의형식에 따른 중복저장 비율이 높아짐에 따라 구조정보의 오버헤드가 높아지게 되었다.

따라서, 본 연구에서는 상이한 계층정보를 동일화하고, 기존의 분리 저장방식을 통합할 수 있도록 관계형 데이터베이스 스키마를 개선, 이를 적용한 저장 모델링 방법을 제안했다. 그 결과 제안 모델링 방법에서는 기존 관계형 데이터베이스 스키마 구조의 개선을 통해 XML 스키마와 인스턴스간 동일한 계층구조 정보를 유지하게 되었으며, 저장 공간 활용 및 비율에 따른 효율성이 높게 되었고, 계층구조 상이에 따른 손상을 최소한으로 유지하게 되어 별도의 연산시간 없이 문서간의 검색질의가 수월하게 되었다. 따라서, 이를 이용한 애플리케이션의 개발시 개발 기간을 단축시킬 수 있을 것이다. 하지만 제안한 저장구조를 적용하면 하나의 엘리먼트에 대한 속성이 여러개 일때, 각 속성에 따른 유효성 체크가 힘들고, XML 문서상의 갱신이 발생하였을때, 갱신에 따른 저장을 고려하지 않았다.

향후 연구는 제안한 구조를 바탕으로 XML 문서자체의 한 엘리먼트에 대한 다수 속성에 따른 유효성 처리 및 해당 속성 검색에 따른 추가 연구 및 갱신에 따른 저장 모델링 방법에 대한 추가 연구가 필요할 것으로 여겨진다.

참고문헌

1) D. W. Shin, 1998, BUS:An Effective Indexing and Retrieval Schema in Structured Documents, in Proc. Digitial Libraries.
2) 권훈, 김정희, 곽호영, 2004, 정적 테이블 기반의 XML 문서 저장 시스템 개선, 한국정보과학회, 제 31권 1호, pp. 178-180.
3) http://www.w3.org/TR/REC-xml, Extensible Markup Language (XML) 1.0 (Second Edition),
4) http://www.w3.org/XML/Schema, XML Schema
5) Toung Dao, 1998, An Indexing Model for Structured Documents to Support Queries on Content, Structured and Attributes, Proceedings of ADL'98, pp.88-97.
6) Brian Lowe, Justin Zobel, Ron Stacks Davis, 1995, A Formal Model for Databases of Structured Text, Proceedings of the Fourth International Conference on Database System for Advanced Applications (DASFAA'95), pp.449-456.
7) 박종관, 이병엽, 손충범, 강형일, 유재수, 2001, XML 문서의 효율적인 구조 검색을 위한 색인 모델, 한국정보처리학회 논문지, 제8-D권, p451-460.

8) 홍석건, 김정희, 곽호영, 2003, 저장과 색인의 효율성을 고려한 정적 테이블 기반의 XML 문서 저장 시스템 설계, 한국정보과학회, 제 30권 2호, pp. 205-207.