RDBMS와 IRS를 이용한 XML 저장관리 시스템 설계 및 구현

(Design and Implementation of a XML Repository System using RDBMS and IRS)

강 형 일 † 최 영 길 †† 이 종 설 † 유 재 수 ††† 조 기 형 †††

(Hyung Il Kang) (Young Gil Choi) (Jong Sul Lee) (Jae Soo Yoo) (Ki Hyung Cho)

요 약 본 논문에서는 관계형 데이타베이스인 오라클과 IRS중 하나인 BRS를 사용하여 XML저장관리 시스템을 설계 및 구현한다. XML저장관리 시스템의 내용 검색과 인덱스 추출을 위해 BRS 검색 시스템을 사용하였으며, XML 문서, 구조정보, DTD, 이미지 등을 저장하기 위해 오라클을 사용하였다. 본 논문에서 구현한 저장관리 시스템은 질의 처리기, 검색결과생성기, XML 객체관리자, XML 인덱스 관리자, 구조검색엔진 등으로 구성된다. 구현된 XML 저장관리 시스템은 XML 문서에 대한 내용검색뿐만 아니라 구조적 특징 또는 애트리뷰트에 기반한 검색을 효율적으로 제공한다. 구현한 저장관리 시스템은 문서 저장 시간, 문서 추출 시간, 내용 검색 시간 등에 대해서 분할 모델 저장관리 시스템과 비교한다.

Abstract In this paper, we design and implement a XML repository system using RDBMS (ORACLE) and IRS(BRS). Our scheme uses BRS to support full text indexing and content-based queries efficiently, and ORACLE to store XML documents, multimedia data, DTD and structure information. In addition, a XML repository system consists of a query processor, a retrieval result generator, a XML object manager, a XML index manager, and a structure retrieval engine. The implemented XML repository system efficiently supports retrieval based on structure and attribute retrieval as well as content-based retrieval. We compare our repository system with decomposition model repository system in terms of document storing time, document extracting time, contents retrieval time.

1. 서 론

최근 인터넷의 발전에 따라 다양한 형태를 가진 정보들이 기하급수적으로 증가되었다. 이에 따라 사용자들은 다양한 형식의 문서에서 정보를 효율적으로 관리 및 공유하고, 구조정보의 표현을 위해 임의로 태그를 생성할 수 있는 기능을 필요로 하게 되었다.

이러한 문서의 논리적 구조를 표현하는 대표적인 표준안으로는 HTML, SGML, XML 등이 있다. XML (eXtensible Markup Language)은 HTML과 SGML의 장점을 수용하여 문서의 논리적인 구조를 표현하면서도 쉽게 사용할 수 있도록 만든 것으로 1998년 W3C에서 XML을 권고안으로 채택하였다[10]. 현재 인터넷 Web 문서, 전자도서관, CSCW, EDI, EC, CALS 등을 포함한 다양한 분야, 수학 분야의 MathML, 채널 기술의 CDF, 이동 통신에서의 WML 등에서 XML 연구는 상당히 활발하게 진행되고 있으며 이러한 분야에서의 대용량의 XML 문서 관리 및 검색을 효율적으로 지원하는 시스템의 필요성이 대두되었다[3].

XML 문서의 효율적인 저장, 관리 및 다양한 검색을 지원하는 XML 저장관리 시스템은 기존의 일반 문서관리 시스템과는 달리 다음과 같은 특성을 고려하여야 한다[6,7,8]. XML 문서는 논리적인 구조와 다양한 멀티미디어를 포함할 수 있다. 이러한 XML 문서는 구조정보

· 본 연구는 (주)한국지식웨어 연구용역과제 지원비에 의해 수행되었음.

† 비 회 원 : 충북대학교 정보통신공학과
khi@pretty.chungbuk.ac.kr
†† 비 회 원 : 한국원자력연구소 환경시스템해석분야 연구원
choiyou@kaeri.re.kr
††† 종신회원 : 충북대학교 정보통신공학과 교수
khjoe@cbucc.chungbuk.ac.kr
yjs@cbucc.chungbuk.ac.kr

논문접수 : 1999년 11월 17일
심사완료 : 2000년 10월 31일

를 이용하여 문서를 효율적으로 관리하기 때문에 데이타베이스에 XML 문서, DTD, 구조정보 및 다양한 미디어를 저장, 관리해야 된다. 또한 기존의 정보검색시스템(IRS: Information Retrieval System)은 문서의 논리적 구조에 따른 정보검색을 거의 이용하지 못하고 있다. 그러나 XML 문서의 특성을 볼 때 XML 문서에 대한 내용검색, XML 문서가 갖는 논리적인 계층 구조를 이용한 검색, 엘리먼트가 갖는 속성에 대한 검색, 혼합검색 등을 수용할 수 있어야 한다.

현재 구현된 대부분의 XML 저장관리 시스템은 DBMS 기반의 시스템을 구현하고 있다. 그러나 DBMS의 안정적이고 저장, 관리 측면의 우수한 성능에도 불구하고 일반적인 IRS에서 제공하는 내용검색에 비해 검색 성능이 현저히 떨어지며, DBMS에서는 일반적으로 길이가 정해지지 않은 문자열에 대해서는 인덱스 생성이 부자연스럽다는 단점이 있다. 반면 기존의 정보검색시스템은 문서의 구조를 표현할 수 없지만 일반적인 내용기반의 검색에서는 DBMS보다 월등한 검색 성능을 보이며, 전문(full-text)에 대한 형태소 분석을 통한 인덱스 생성이 DBMS에 비해 탁월하다[11]. 때문에 DBMS만의 기능에 의존한 XML 저장관리 시스템은 XML 문서의 특성을 고려한 저장, 관리 및 검색을 지원하는데 한계점이 있다.

이에 본 논문에서는 대용량 XML 문서의 효과적인 저장과 검색을 위하여 DBMS와 IRS의 장점만을 이용하여 XML 저장관리 시스템을 설계하고 구현한다. 이를 위해 현재 가장 많이 쓰이고 있는 관계형 데이타베이스인 오라클을 저장시스템으로 활용하고 인덱스 생성 및 키워드 검색을 위해 BRS 검색엔진을 이용한다. 관계형 데이타모델은 현재 널리 사용되는 관계형 데이타베이스를 사용함으로써 데이타베이스의 활용이 쉬우며 상용 검색엔진인 BRS를 사용함으로써 데이타베이스의 키워드 검색능력보다 탁월한 검색능력을 보일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 XML 저장관리 시스템의 기존 연구 동향에 대해 살펴보고 3장에서는 설계한 XML 구조정보 및 모델링을 설명한다. 4장에서는 구현한 XML 저장관리 시스템의 구조 및 각 모듈에 대해 알아본다. 5장에서는 구현한 시스템의 성능평가를 수행한다. 6장에서는 마지막으로 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

기존의 구조문서를 저장, 관리 및 검색을 지원하는 문서관리시스템의 연구는 주로 DBMS를 활용하는 방법을 사용한다. 구조 문서의 논리적인 구조적 특성을 다루기 위한 저장 시스템으로서 기존의 DBMS를 사용하는 것은 효율적인 적용에 한계가 있지만, 다른 정보 저장시스템에 비해 안정적이고 DBMS의 다양한 기능을 활용할 수 있고, 또한 이미 널리 사용되고 있는 장점이 있다. 이러한 이유로 인해 구조 문서의 구조정보와 함께 문서를 분할 모델, 비분할 모델, 혼합 모델로 저장하고 다양한 검색을 지원하는 방법에 대한 연구가 진행되어 왔다[1,2,4,5,9]. 분할 모델이란 XML 문서를 저장할 때 문서를 구성하고 있는 엘리먼트를 나누어 저장하는 방법으로 해당 문서에 대한 검색이 발생한 경우 엘리먼트들을 재구성하여 검색 결과를 반환하는 과정에서 시스템의 성능을 저하시키는 문제가 발생한다. 비분할 모델은 XML 문서 전체를 저장한 후 각각의 엘리먼트는 위치정보(시작위치, 끝나는 위치, 길이)를 가지고 접근하는 방식이다. 이는 문서를 한꺼번에 저장하였기 때문에 통합 과정이 필요 없어 문서 참조를 빨리 할 수 있지만, 내용의 일부만이 수정되었을 때도 문서 전체를 재구성해야 한다는 단점이 있다. 혼합모델은 분할 저장 모델과 비분할 저장 모델을 혼용하여 사용하는 모델로 각각의 모델에서 단점을 보완하고자 상대 모델의 특성을 일부 포함하였다. 하지만 혼합 모델의 단점인 저장 공간이 많이 소모된다는 문제점이 있다.

XML 문서의 저장 및 다양한 검색을 지원하기 위해 오라클 8i에 기반한 검색 시스템이 개발되었다[2]. 이 연구에서는 XPointer를 기반으로 하여 질의 언어를 설계하였고, XML이 지닌 내용, 구조, 애트리뷰트에 대한 검색뿐만 링크 정보에 대한 검색도 지원하였다. 그러나 SGML 문서 내에 포함되는 멀티미디어의 저장에 대한 고려가 없다.

RMIT에서는 GCL 구조를 확장한 SCL 모델을 제안하였는데, 이 방법은 모든 DTD를 수용할 수 있는 모델링을 제공한다[14]. 이 모델은 SGML 문서내의 텍스트와 마크업에 대해 색인 넘버를 부여한 후 이들의 텍스트 간격과 포함관계를 사용하는 방법으로 불용어들을 제외한 텍스트 어휘들은 텍스트 인덱스에 색인넘버로서 저장되고, 마크업은 시작태그와 마침태그의 쌍으로 마크업 인덱스에 저장된다. 이 연구에서는 내용검색, 단순한 포함관계의 구조검색, 애트리뷰트 검색 등을 지원한다. 그러나 SCL 구조는 색인어가 포함된 엘리먼트에 대해 트리내의 깊이를 표현할 수 없다는 단점을 가진다. 즉, 특정 엘리먼트의 조상, 형제를 알 수 없으며 포함하고 있는 엘리먼트가 몇번째 자손에 해당하는지 알 수 없다.

Syracuse 대학에서는 SGML 문서의 저장, 관리 및

검색을 지원하기 위해 OODBMS를 활용한 효율적인 데이타 모델링, 색인 구조 및 질의어를 제안하였다[9]. 제안하는 방법은 문서 내에 포함될 수 있는 멀티미디어에 대한 저장 및 관리를 제공하며, 내용 검색 뿐만 아니라 구조, 애트리뷰트, 혼합 검색 등을 지원할 수 있다. 이와 같은 다양한 검색을 지원하기 위해 제안한 구조정보는 lid(logical object UID) 이외에 문서의 레이아웃 정보를 이용한 sid(spatial layout object UID)와 tid(temporal layout object UID) 세 개의 정보로 구성이 된다. 그러나 레이아웃에 관련된 UID를 구하기 위해서는 해당 SGML 문서에 반드시 포맷터를 적용하여 포맷팅을 하여야 한다. 즉, DTD를 적용하여 생성한 SGML 문서만을 가지고는 이 방법을 적용할 수 없다.

또한 SGML 문서의 다양한 검색을 지원하는 색인 구조로 K-ary를 이용하는 방법을 제안하였다[1]. 이 연구에서 제안한 구조 정보는 SGML 문서를 구문 트리로 구성한 후 가장 큰 자식의 수 k를 구하여 해당 문서를 K-ary 재구성한 후 문서의 구문 트리를 매핑하여 각 노드에 노드 번호를 부여하는 단점이 있다.

SGML 문서에 대한 내용과 구조가 혼합된 검색에 초점을 맞춘 추론망에 기반한 새로운 모델이 개발되고 구현되었다[13]. 이 연구에서는 SGML 문서의 구조정보를 적절히 이용할 경우 문서 단위 검색에 있어서의 신뢰도 향상에도 큰 영향을 줄 수 있음을 보였다. 그러나 단순 구조 검색 및 애트리뷰트 검색은 고려하지 않았다.

GMD에서는 SGML 문서를 저장하고 검색하기 위하여 OODBMS와 검색 시스템을 통합하는 방법을 제공하고 내용 질의와 구조 질의를 OODBMS의 질의어로 표현할 수 있도록 하였다. 하지만 이 방법은 검색 시 사용자의 질의어와 각 단위 사이의 유사도 계산뿐만 아니라 질의어 위치와의 관계를 기록하여야 하는 부담이 있다[12].

3. XML 저장관리 시스템 설계

3.1 XML 구조정보

기존의 구조정보 표현 연구들은 특정 엘리먼트에 대한 직접적인 접근이 불가능하고 조상, 자손, 형제의 관계에 있는 엘리먼트를 접근하기 위해 복잡한 연산을 수행해야 한다. 이에 본 논문에서는 특정 엘리먼트에 대한 직접적인 접근이 가능하고 엘리먼트 간의 관계를 구하기 위해 복잡한 연산이 필요 없도록 DTD의 논리적 구조정보를 사용해서 XML 문서를 효율적으로 관리할 수 있는 구조정보를 제안한다.

이를 위해 XML 문서의 구조정보를 표현하기 위해

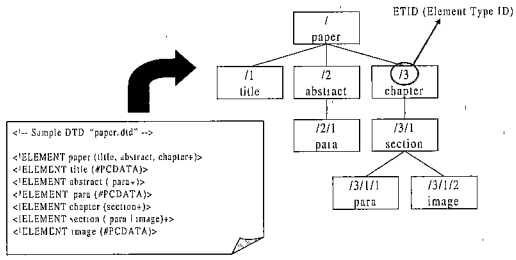

그림 1 ETID의 할당

ETID(element type ID), SORD(sibling order), SSORD(same sibling order)를 고안하였다. ETID (element type ID)는 XML문서의 논리적 구조를 이루는 DTD의 각 엘리먼트에 할당되는 유일한 값으로 문서 구조를 트리 형태로 표현하여, UNIX 파일 시스템에서 디렉토리를 표현하는 방법과 같은 방법을 이용하여 부여한다. 즉 root 엘리먼트는 '/'로 표현되며 root 엘리먼트에 포함되는 엘리먼트는 '/1', '/2'와 같이 표현된다. 다음 (그림 1)은 문서구조를 트리로 표현하고 각 엘리먼트에 ETID를 할당한 예이다.

엘리먼트간의 형제 노드들간의 순서정보(SORD : sibling order)와 동일 타입의 엘리먼트들 간의 순서정보(SSORD : same sibling order)의 표현은 XML 문서 인스턴스에 적용한다. 형제 노드들간의 순서정보(SORD)는 동일 부모를 갖는 엘리먼트들의 출현 순서이며, 동일 부모를 갖는 엘리먼트들 중 동일한 형(type)간의 순서는 SSORD로 표현된다. 자식 엘리먼트의 순서정보에는 부모 엘리먼트의 순서정보도 함께 표현되는데 표현방법은 ETID의 표현방법과 동일하다. 다음 (그림 2)는 XML 문서의 구조정보를 본 논문에서 제안하는 방법으로 트리 형태로 표현한 예이다.

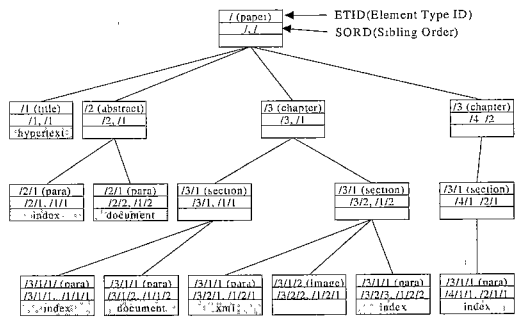

그림 2 XML 문서의 구조정보

3.2 스키마 설계

XML 문서는 복잡한 구조와 다양한 미디어를 포함하기 때문에 데이타베이스에 XML 문서, DTD, 구조정보 및 다양한 미디어 등을 저장, 관리해야 된다. 이를 위해 XML 데이타 모델링이 필요하며 본 논문에서는 하부 저장 시스템으로 사용하는 ORACLE 기반의 관계형 모델을 사용하였으며, XML 문서를 저장함에 있어서 문서의 빠른 추출을 위하여 비분할 저장 모델을 고려하였다. 본 논문에서 구현한 XML 저장관리 시스템을 위한 스키마 구조는 (그림 3)과 같다.

DTD 테이블 : dtd_table

dtdid	number
dtdname	char
content	varchar2

문서 테이블 : DTD명_doc

did	number
name	char
fulltext	long

엘리먼트 테이블 : DTD명_node

did	number
sord	char
S_offset	number
length	number

이미지 테이블 : DTD명_image

did	number
name	char
S_offset	number
E_offset	number
content	long raw

그림 3 스키마 구조

DTD 테이블은 저장할 XML 문서의 DTD를 저장, 관리한다. 새로 저장되는 DTD는 오라클의 long type으로 저장이 된다. XML문서 요구시 해당 DTD의 dtdid 또는 DTD명을 통해 제공된다. 문서 테이블은 저장되는 XML문서는 문서 번호와 문서이름을 가지고 XML 문서 전체를 저장, 관리한다. XML문서는 long type으로 저장되며 did 나 문서명을 통해 제공된다. 엘리먼트 테이블은 구조정보 추출기를 통해 추출된 구조정보 중에서 XML 문서의 각 엘리먼트의 구조정보를 저장, 관리한다. 해당 엘리먼트에 대해 특정 XML문서안에서의 위치를 나타내기 위해 문서번호, sord, 문서안에서의 시작위치, 엘리먼트의 전체 길이를 저장한다. 이미지 테이블은 XML 문서에 포함되어 있는 이미지 저장을 위해 이미지가 정의된 애트리뷰트의 문서번호, 이미지 파일명, XML 문서 안에서의 시작위치와 끝나는 위치정보를 저장한다.

3.3 XML 질의 분석

XML 문서를 위한 질의는 내용 검색과 구조 검색, 애트리뷰트 검색, 그리고 내용이나 구조, 애트리뷰트가 복합된 검색이 있다.

내용 검색은 사용자에 의해 주어지는 키워드와 관련된 문서나 엘리먼트를 검색하는 질의이다.

- 질의 1 : "데이타베이스"를 포함하는 문서를 찾아라.

구조 검색은 문서의 논리적인 구조에 기반한 질의로서, 엘리먼트들의 계층간의 관계, 같은 계층내에서의 관계, 계층 전체 관계 등을 고려하여야 한다. 계층간의 관계는 부모/자식/조상/자손 관계가 있으며, 같은 계층내의 관계는 형제 엘리먼트간의 순서가 있다. 그리고 계층 전체 관계는 선후관계가 있다. 그러므로 이와 같은 구조 관계에 의해 원하는 엘리먼트를 지정할 수 있어야 한다.

- 질의 2 : 「TITLE」 엘리먼트의 부모 엘리먼트를 찾아라.(계층간의 관계 질의)

- 질의 3 : 「TITLE」 엘리먼트의 자식들 중 세 번째 「CHAPTER」를 찾아라.(같은 계층내의 관계 질의)

- 질의 4 : 「TITLE」 엘리먼트의 형제들 이전에 나오는 엘리먼트를 찾아라.(계층 전체 관계 질의)

애트리뷰트 검색은 엘리먼트에 나타날 수 있는 속성에 질의로 애트리뷰트 이름과 값을 주고 해당하는 문서나 엘리먼트를 찾는 질의이다.

- 질의 5 : 애트리뷰트 「SEX」가 MALE인 문서를 찾아라.

혼합 검색은 엘리먼트의 내용, 구조기반, 속성에 대한 검색이 혼합되어 사용되는 질의이다.

- 질의 6 : 「TITLE」 엘리먼트에 "XML"를 포함하는 문서를 찾아라.

- 질의 7 : 「PARA」와 「SECTION」을 하위 엘리먼트로 갖는 엘리먼트 중 "구조정보"를 포함하는 엘리먼트를 찾아라.

- 질의 8 : 「TITLE」 엘리먼트의 자식들 중 "XML"를 포함하고, 애트리뷰트 「YEAR」가 "1999"인 문서를 찾아라.

4. XML 저장관리 시스템 구현

본 논문에서 구현한 XML 저장관리 시스템은 하부 저장 시스템으로 ORACLE 7.3을 사용하였으며, ORACLE에 비해 일반적인 키워드 검색이 뛰어나며, 길이가 정해지지 않는 문자열에 대한 인덱스 생성을 위해 BRS 검색엔진을 활용하였다. (그림 4)는 구현한 XML 저장관리 시스템 구조를 보여 준다. XML 저장관리 시스템은 질의처리 및 검색결과 생성기, XML 객체 관리자, XML 인덱스 관리자, BRS 검색엔진, 구조검색처리기로 구성된다. XML 저장관리 시스템을 구성하는 각 모듈의 역할은 다음과 같다.

XML 저장관리 시스템에서 가장 핵심적인 기능을 담당하는 XML 객체관리자에서는 실제 XML 문서를 저장하기 위한 스키마 생성 및 XML 문서 인스턴스의 저

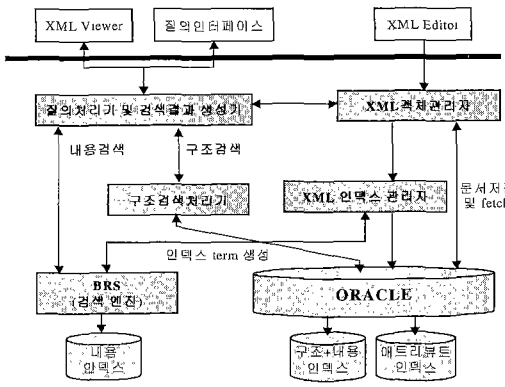

그림 4 XML 저장관리기의 시스템 구성도

장 및 추출을 담당한다. XML 인덱스관리자는 구조검색, 애트리뷰트검색, 혼합검색 등을 처리하는 인덱스를 ORACLE 기반으로 생성하고 관리한다. BRS 검색엔진은 내용인덱스를 생성, 관리하며 사용자의 질의 중 키워드 검색을 처리한다. 구조검색처리기는 BRS에서 처리하지 못하는 구조검색, 애트리뷰트검색, 혼합검색을 처리한다. 질의처리기에서는 사용자 질의를 분석하여 내용검색은 BRS 검색엔진이 처리하고, 구조검색, 애트리뷰트검색, 혼합검색은 구조검색처리기를 사용한다. 또한 검색결과생성기에서는 BRS와 구조검색엔진이 처리한 검색 결과를 이용하여 문서 전체 또는 일부분을 사용자에게 제공한다.

4.1 XML 객체관리자

XML 객체관리자는 XML 문서를 효율적으로 관리하기 위한 문서의 일관된 관리 기능을 제공한다. 이를 위해 먼저 XML 객체관리자는 3장에서 설계한 XML 문서의 스키마를 데이타베이스에 생성한다. 또한 실제 XML 문서, DTD, 구조정보, 이미지 등을 데이타베이스에 저장하며, 저장된 XML 문서를 사용자가 원하는 전체 문서 또는 문서 일부분을 꺼내는 일을 담당한다.

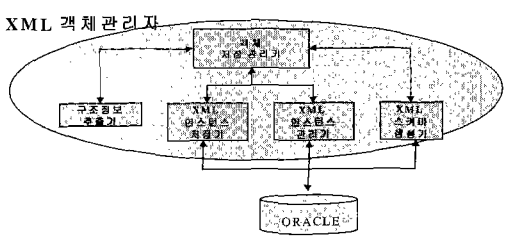

그림 5 객체관리자의 구성도

Tag name	DID	ETID	SORD	SSORD	Type	Start offset	End offset	Content length	Content

그림 6 구조 정보추출 결과

XML 객체관리자를 구성하는 세부 모듈은 (그림 5)와 같이 객체 저장관리기, 구조정보 추출기, XML 인스턴스 관리기, XML 인스턴스 저장기, XML 스키마 생성기 등으로 이루어진다. 객체 저장관리기는 XML 객체관리자를 구성하는 각 모듈들에 대한 통합 인터페이스를 제공한다. 구조정보 추출기는 XML 인스턴스에서 문서를 저장하는데 필요한 구조정보를 추출하는 역할을 한다. XML 구조정보는 ETID, SORD, SSORD등으로 구성되며, 구조정보 추출기에 의해 추출되는 정보는 (그림 6)과 같다.

- Tag name : 엘리먼트 이름 또는 애트리뷰트 이름
- DID : 문서의 고유번호
- ETID : 엘리먼트 형(type) ID
- SORD : 형제 노드들간의 순서정보.
- SSORD : 동일타입 엘리먼트의 순서정보.
- Type : 엘리먼트인지 애트리뷰트인지를 나타냄
- Start_offset : 엘리먼트 및 애트리뷰트의 시작 위치
- End_offset : 엘리먼트 및 애트리뷰트의 끝 위치
- Content : 엘리먼트에 포함되는 PCDATA 또는 애트리뷰트일 경우 해당값

XML 인스턴스 저장기는 DTD, XML 문서, 구조정보, 이미지 등을 데이타베이스에 저장하기 위한 모듈이다. 저장할 DTD 와 XML문서는 해당 DTDID 와 DID를 할당한 뒤 문서 전체를 저장하며, 문서의 구조정보는 구조정보추출기의 결과를 바탕으로 엘리먼트에 대해서 DID, SORD, Start_offset, length를 저장하게 된다. XML 문서 내에 애트리뷰트로 정의되어 포함된 이미지는 해당 애트리뷰트의 구조정보(DID, 파일명, Start-offset, End_offset)와 함께 저장된다. 다음 (그림 7)은 XML 문서의 저장 과정을 보여준다.

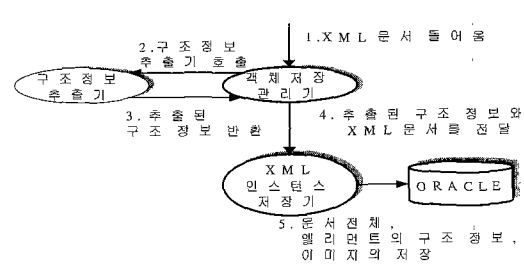

그림 7 XML 문서의 저장 과정

이미지에 대한 저장은 오라클의 long raw type 을 사용하여 저장하였다. 이미지 테이블에 대한 스키마는 (그림 3)과 같다. 실제 XML 문서 내에서 이미지를 기술한 부분과 구조정보 추출기를 통해서 추출된 결과에 대한 예제는 (그림 8)과 같다. 이미지를 나타내는 <figGrp> 엘리먼트는 실제 XML 문서 안에서의 시작 위치는 21666에서 끝 위치는 21958이고 이 엘리먼트의 애트리뷰트인 name에서 fig1.gif 라는 이미지 파일을 기술하고 있다. 이미지 테이블에는 해당 이미지가 포함된 문서의 DID, 파일명, start_offset, end_offset, content 가 저장되는데 그 값은 (1, fig1.gif, 21741, 21765, 이미지 내용)이 된다.

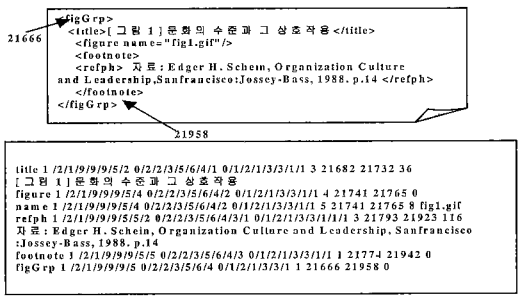

그림 8 이미지 기술에 대한 구조정보 표현

XML 인스턴스 관리기는 사용자가 요구하는 문서 전체 혹은 문서 일부분과 포함된 이미지를 데이타베이스로부터 추출하는 기능을 담당한다. XML 인스턴스 관리기는 (그림 9)에서와 같이 문서 전체에 대해서는 해당 문서에 대한 DID나 문서 명을 통해서 추출이 된다. 문서의 일부분인 엘리먼트에 대해서는 해당 문서의 DID와 해당 엘리먼트의 SORD를 통해서 문서 내에서의 위치정보(Start_offset, length)를 구하고 해당 위치정보를 바탕으로 문서 전체에서 일부분을 추출한다. 구조정보와 함께 저장된 이미지는 추출되는 엘리먼트의 구조정보를 바탕으로 해당 엘리먼트가 포함하는 모든 이미지를 데이타베이스에서 추출하여 제공한다. 문서 전체에 대한 추출 시 인스턴스 관리기는 해당 문서의 DID를 가지는 이미지를 추출한다. 문서 일부분에 대한 추출 시는 추출되는 엘리먼트의 start_offset과 length를 바탕으로 이미지의 start_offset이 엘리먼트의 start-offset보다 크고 이미지의 end_offset이 엘리먼트의 end-offset보다 작은 이미지에 대해서 추출한다. (그림 8)에서 이미지를 포함하고 있는 figGrp가 추출될 경우에 figGrp의 인스턴스에서의 위치는 21666에서 21958이 된다. 이때 21666에서 21958안에 기술되어 있는 이미지를 추출하면 해당되는 이미지를 가져올 수 있다. (그림 8)에서는 fig1.gif 만이 추출되어 진다.

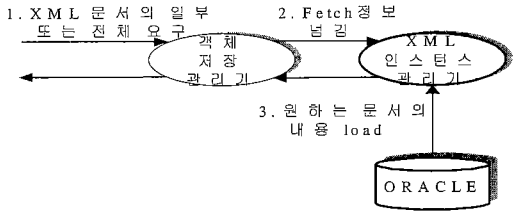

그림 9 XML 문서의 fetch 과정

XML 스키마 생성기는 DTD 테이블을 시스템 초기화에 생성하고 이후 다양한 DTD를 수용할 때마다 문서테이블, 엘리먼트 테이블, 이미지 테이블을 동적으로 생성한다. 각 테이블은 DTD명에 따라서 DTD명_doc, DTD명_node, DTD명_image 로써 생성된다. 스키마의 생성과정은 (그림 10)과 같다.

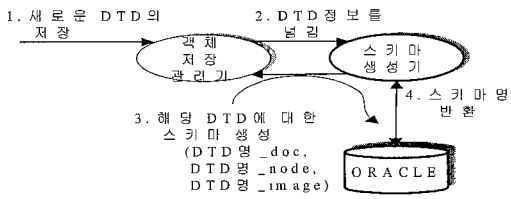

그림 10 스키마 생성과정

4.2 XML 인덱스관리자

XML 문서에 대해 내용, 구조, 애트리뷰트, 혼합검색을 지원하기 위한 색인구조를 만들고 색인 정보를 관리하는 모듈이 XML 인덱스 관리자이다. XML 인덱스 관리자에서는 키워드 검색을 위한 내용 색인기, 구조 검색과 구조와 내용을 동시에 지원하는 XML 구조+내용 색인기, 애트리뷰트 검색을 지원하기 위한 애트리뷰트 색인기로 구성된다.

내용 색인기에서는 빠른 full-text 검색을 위해 BRS 검색엔진의 색인을 이용하고 검색 결과로서 실제 문서는 BRS 검색엔진에서 부여한 식별자와 XML 객체관리자에서 부여한 식별자 정보를 유지하는 매핑 정보를 이용하여 사용자에게 보내진다. (그림 11)은 내용 인덱스의 생성 과정을 보여준다. 먼저 XML 인덱스관리자에서

색인 할 문서를 전달받으면 DID를 추출하고, XML 문서에서 태그 부분을 삭제하여 인덱싱 과정을 거쳐 BRS 검색엔진에서 검색에 사용되는 문서식별자(DOCN)와 XML 객체 관리기에서 사용되는 문서식별자(DID)를 매핑 할 수 있는 매핑 모듈이 필요하다. 매핑 정보는 오라클에 저장하여 관리한다.

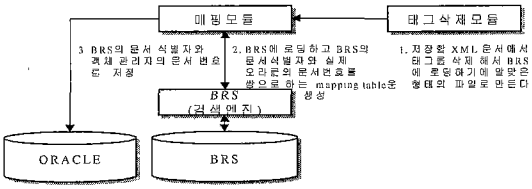

그림 11 내용 검색 인덱스 생성 과정

(그림 12)는 본 논문에서 설계한 구조+내용 색인 구조이다. 그림에서 보듯이 단순한 구조질의와 구조+ 내용이 혼합된 질의를 빠르게 처리하기 위해 인덱스 화일에서는 엘리먼트 식별자인 ETID와 키워드를 가지고 인덱싱한다. 이렇게 함으로서 엘리먼트 기반의 내용 검색을 빠르게 지원할 수 있다. 포스팅 화일에서는 검색 결과로서 해당하는 특정한 엘리먼트나 문서를 접근하기 위한 정보들로 구성된다. 특히 이 정보들 중에 SORD와 SSORD를 이용하여 여러 XML 문서에서 고유한 엘리먼트를 접근할 수 있다.

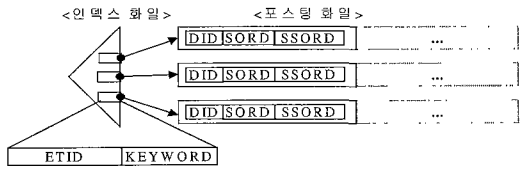

그림 12 구조+내용 색인구조

(그림 13)은 애트리뷰트 검색을 지원하기 위한 색인 구조이다.

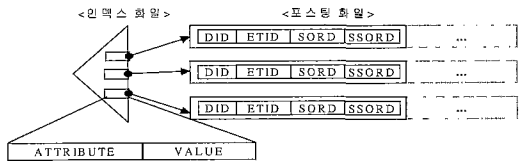

그림 13 애트리뷰트색인 구조

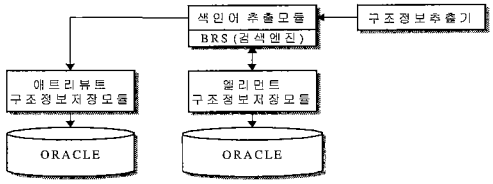

그림 14 구조+내용 색인 및 애트리뷰트 색인 과정

위와 같은 색인구조는 (그림 14)와 같은 과정을 통해 구조+내용 색인 및 애트리뷰트 색인을 생성한다. 먼저 XML 문서를 저장하면서 할당받은 DID를 문서에 포함하여 구조정보 추출기에 넘기면 구조정보 추출기는 (그림6)에서 보여주는 형식과 같이 해당 문서에 대한 구조정보를 추출하여 구조정보 파일을 생성한다. 생성한 파일을 BRS 검색엔진의 색인어 추출모듈에 넘기고 색인어 추출모듈은 구조정보 파일의 각 필드들을 읽고 여기서 구조정보 색인을 만들기 위해 필요한 정보만을 뽑아낸다. 엘리먼트를 색인하는 경우 엘리먼트의 구조정보를 DID, ETID, SORD, SSORD로 구성된 구조체의 링크드 리스트로 생성하고 이와 함께 엘리먼트의 구조정보 색인을 구성하는 색인어는 BRS를 통하여 추출하게 된다. 애트리뷰트를 색인하는 경우 애트리뷰트 이름, 애트리뷰트 값, ETID, SORD, SSORD, DID로 구성된다. 이렇게 구성한 각 색인 정보들은 ORACLE에 저장, 관리되며 내용검색 이외의 다양한 검색은 구조+내용 색인 및 애트리뷰트 색인을 참조하여 처리된다.

4.3 XML 질의 처리기 및 검색결과 생성기

질의처리 및 검색결과 생성기는 내용검색과 구조검색을 구분하여 내용검색이 사용자로부터 요청될 경우는 BRS 검색엔진에서 처리하고 처리된 결과는 검색 결과 생성기에서 XML 객체관리자의 XML 인스턴스관리기를 통해 ORACLE에 저장되어 있을 문서전체 혹은 일부분을 사용자에게 보여 준다. 구조검색, 애트리뷰트검색, 혼합검색이 요청될 경우는 구조검색처리기가 검색한 결과를 이용하여 문서 전체 또는 일부분을 사용자에게 제공한다.

사용자가 질의 인터페이스를 통해 질의를 내릴 경우 질의처리기는 질의를 분석하여 내용 검색을 위한 질의일 경우 처리 과정은 (그림 15)과 같다.

내용 검색 모듈은 크게 두 모듈 BRS DOCN 검색모듈과 오라클 DID 검색 모듈로 나눌 수 있다. 먼저 BRS DOCN 검색 모듈은 XML 문서에 대해 내용 검색을 할 때 사용되며 주로 BRS API를 이용하여 구성되어 있다. 이 모듈의 전체적인 흐름은 우선 BRS 검색엔진을 시작하고 BRS 자체에서 사용하는 화일 시스템 기반의 DB

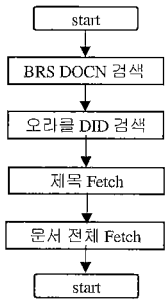

그림 15 내용 검색 모듈

를 선택한 다음 BRS 엔진의 검색 파라미터를 설정한다. 주어진 검색어를 탐색하여 검색결과에 대해 back reference 번호를 가지고 검색된 문서들의 DOCN을 링크드 리스트로 생성하여 오라클 DID 검색 모듈로 전달한다.

오라클 DID 검색 모듈은 전달받은 DOCN 결과를 가지고 오라클 데이타베이스에 있는 DOCN_TABLE을 검색하여 DOCN에 해당하는 DID를 검색하여 링크드 리스트로 생성하고 각 DID에 해당하는 문서들의 제목을 출력하고 해당 문서를 선택하면 전체 문서를 출력한다.

4.4 구조질의처리기

내용검색이외의 다양한 검색은 3장에서 분석한 질의를 모두 수용할 수 있도록 사용자 인터페이스를 구성하고 구조검색은 부모/자식/조상/자손/형제 등 크게 5개의 질의타입으로 분류하고 각각에 대하여 해당 API들을 적용함으로써 이루어진다. 애트리뷰트 질의 처리는 사용자 인터페이스의 입력상태를 분석하여 4개의 질의타입으로 분류하고 각각에 대하여 API들을 적용함으로써 이루어진다. 각 API들은 해당 질의에 대한 SQL로의 변환 모듈로 구성된다.

다음 (그림 16)은 "이동전화를 포함하는 첫 번째 ThesisTitle의 부모 엘리먼트를 찾아라"라는 구조+내용질의 처리과정을 보여준다. 우선 검색의 기준인 시작 엘리먼트의 ETID를 엘리먼트이름-ETID 매핑테이블을 조사하여 구한다. 이것과 키워드인 "이동전화"를 key로 구조+내용 색인을 검색하여 ThesisTitle이면서 "이동전화"를 갖는 문서 엘리먼트의 DID, SORD, SSORD 쌍들을 구한다. 추출된 결과들 중에 첫 번째 ThesisTitle을 추출하기 동일한 형(type)간의 순서 정보를 유지하는 SSORD로부터 SSORD가 "/*/*/1"인 쌍들만 추출한다. 추출된 DID, SORD, SSORD에서 부모 엘리먼트를 구하기 위해 SORD 정보를 이용하여 구한다. 예를 들어 SORD가 "/2/3/1"일 경우 마지막 "/1"을 제거한 "/2/3"이 부모 엘리먼트의 SORD가 된다. 마지막으로 부모 엘리먼트의 SORD와 DID를 이용하여 실질적인 문서 엘리먼트을 구한다.

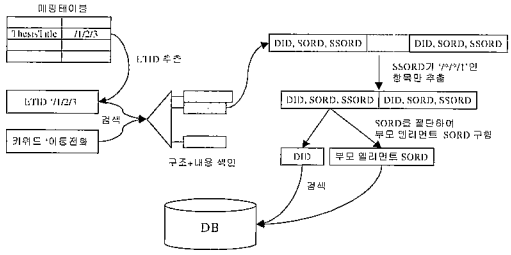

그림 16 구조+내용질의 처리과정

4.5 질의 인터페이스

내용검색은 XML에 표현되어 있는 텍스트에 대한 스트링 매칭 검색을 하고, 구조검색에서는 엘리먼트들의 계층 관계를 위해 조상, 자손의 관계가 있으며, 같은 계층내의 관계를 위해 형제 관계가 있다. 또한 각 엘리먼트들 간의 순서에 따라 선후 관계를 "+, -, 숫자"로 표현할 수 있도록 하였다. 애트리뷰트 검색은 애트리뷰트 값의 다양한 관계 연산을 제공한다. 검색 결과로는 문서 전체 또는 엘리먼트를 지정할 수 있도록 하였다. (그림 17)은 설계한 질의 인터페이스를 보여주고 있으며, "이동전화"를 포함하는 첫 번째 ThesisTitle의 부모 엘리먼트를 찾아라」라는 구조+내용 질의의 예를 보여주고 있다.

먼저 첫 번째 ThesisTitle에 포함된 키워드가 "이동전화"이므로 시작 엘리먼트의 순위값은 '1'이며 엘리먼트는 ThesisTitle, 키워드는 "이동전화"라고 입력한다. 시작 엘리먼트와 찾고자 하는 엘리먼트의 관계가 부모이므로 관계순위는 '1', 관계값은 '조상'으로 한다. 관계

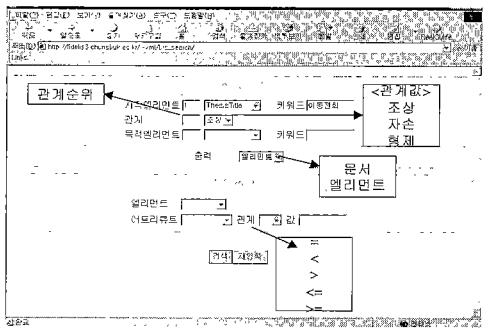

그림 17 구조+내용 질의 인터페이스

표 1 엘리먼트의 관계와 관계값

시작엘리먼트와 목적엘리먼트의 관계	관계값	관계순위
조상	조상	없음
부모	조상	1
자손	자손	없음
자식	자손	1
바로 앞 형제	형제	-1
앞 형제	형제	-
바로 뒤 형제	형제	+1
뒤 형제	형제	+

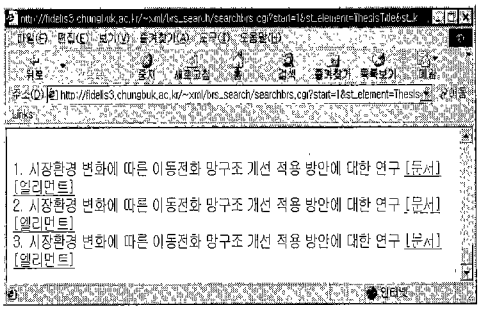

그림 18 구조+내용 질의 결과

값과 엘리먼트간의 관계는 <표 1>과 같다. 출력 방법은 엘리먼트 단위와 해당 문서 전체에 대해서 출력이 가능하다. 다음 (그림 18)은 (그림 17)의 질의에 대한 처리 결과이다.

5. 성능 평가

본 논문에서 구현한 XML 저장관리 시스템의 성능평가는 논문지의 형식에 맞게 설계한 paper.dtd를 가지고 총 70M 분량의 문서 300개를 가지고 SUN Enterprise 3000에서 저장 테스트한 결과이다. 각 문서를 구조에 따라 나누었을 때 294,664개의 엘리먼트가 생성되었다. 또한 각 XML 문서 당 이미지의 평균 개수는 13개, 평균 크기는 37.5K이다. 성능평가는 DBMS 기반의 분할모델과 비교 수행하였다.

(그림 19)는 문서 크기에 따른 저장 시간을 나타낸다. 300개의 문서에 대한 분할모델의 평균 삽입시간은 59.42초이나 본 논문에서 구현한 시스템의 평균 삽입시간은 28.71초로 상당한 저장 속도 향상을 이루었다. 이는 모든 DTD를 수용할 수 있는 스키마를 사용한 점과 비분할 모델을 사용한 점으로 인한 것이다.

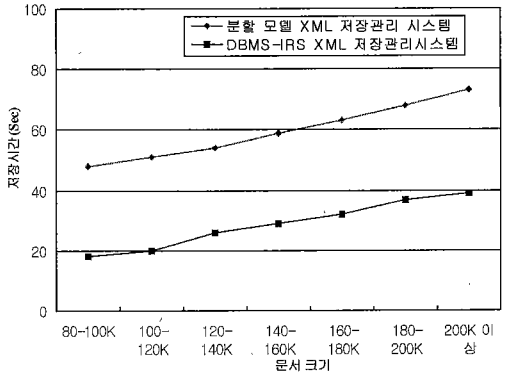

그림 19 XML 문서 저장 시간

(그림 20)은 문서 크기에 따른 추출시간을 나타낸 것이다. 분할 모델의 경우 해당 문서를 구성하는 모든 엘리먼트를 재구성하여 추출하기 때문에 상당히 큰 검색시간을 가진다. 그러나 구현한 방법은 비분할 모델을 사용하기 때문에 상당히 빠른 추출시간을 가진다

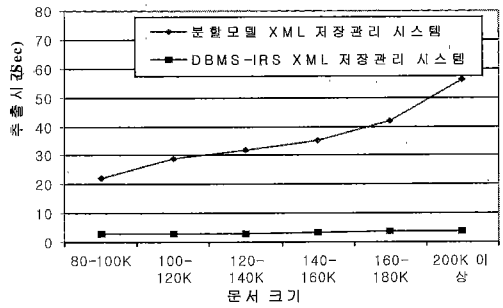

그림 20 XML 문서 추출 시간

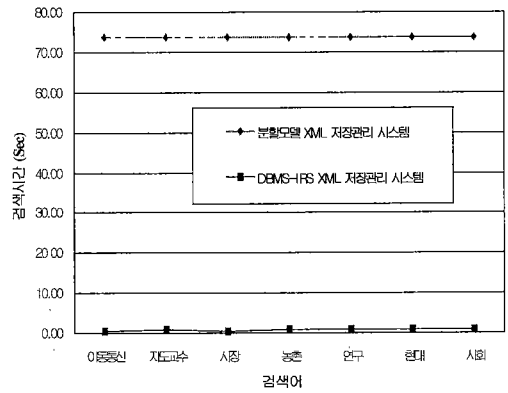

그림 21 내용 검색 시간

내용검색의 경우에는 '이동통신', '지도교수', '시장', '농촌', '연구', '현대', '사회' 등 7개의 키워드를 사용하여 각 키워드에 대해 검색시간을 측정하였다. (그림 21)에서 보는 바와 같이 분할모델은 제안한 방법보다 매우 좋지 않은 성능을 보였다. 이는 일반적으로 DBMS에서 가변길이 문자열에 대한 인덱스를 생성할 수 없기 때문이다. 이 성능평가에서는 약 만개의 엘리먼트가 내용을 저장하고 있는데, 저장하려는 문서의 수가 증가하면 검색성능이 더욱 떨어질 것이다. 구현한 시스템에서는 내용검색을 역화일 기법을 사용하는 검색엔진인 BRS를 사용함으로써 분할모델에 비해 월등하게 성능이 좋았다.

6. 결 론

본 논문에서는 대용량의 XML 문서 관리 시스템을 개발하는데 있어서 DBMS와 IRS의 장점만을 이용한 XML 저장관리 시스템을 설계 및 구현하였다. 구현한 XML 저장관리 시스템의 특징은 XML 문서의 저장, 관리를 위하여 DBMS를 사용함으로써 다른 정보 저장시스템에 비해 안정적이고 DBMS의 다양한 기능을 활용할 수 있으며, 또한 IRS보다 비효율적인 내용검색과 인덱스 생성에 BRS 검색엔진을 활용함으로써 전체 시스템의 성능을 향상할 수 있음을 성능평가를 통해 알 수 있었다. DBMS는 ORACLE을 사용한 관계형 모델이고, 문서 추출의 성능향상을 위해 XML 문서의 전체를 저장하는 비분할 모델을 사용하였으며, 모든 DTD를 수용할 수 있도록 스키마를 설계하였다. 또한 XML 문서 내에 포함되어 있는 멀티미디어의 저장 및 관리에 대한 고려를 하였다. 분할 모델과의 성능평가를 수행한 결과 약 200% 정도의 저장 속도 향상을 가졌으며, 문서 추출 시간은 약 1100% 정도의 성능 향상을 가져왔다.

또한 본 논문에서 XML 문서의 논리적인 구조를 효율적으로 표현할 수 있는 구조정보를 제안하였다. DTD의 각 엘리먼트에 할당되는 유일한 값인 ETID를 사용함으로써 특정 엘리먼트를 직접 검색할 수 있으며, 엘리먼트간의 형제 노드들간의 순서정보인 SORD와 동일 타입의 엘리먼트들 간의 순서정보인 SSORD를 사용함으로써 XML이 내포하는 구조에 맞는 부모/자식/조상/자손 등의 관계 및 형제간의 순서관계, 선후관계 등의 질의를 효율적으로 수행할 수 있었다.

결론적으로 XML 문서의 효율적인 저장, 관리 및 검색을 위하여 DBMS만을 이용한 시스템보다는 IRS의 특징을 활용하는 시스템이 바람직하며, XML 문서의 저장 및 추출 속도 향상을 위해서는 비분할 모델이 우수함을 알 수 있었다. 향후연구과제로는 기존의 구조화된 문서 관리시스템과의 성능평가를 수행하는 것이다.

참 고 문 헌

[1] 손정한, 이회주, 장재우, 심부성, 주종철, "SGML 정보검색을 위한 인덱스 관리자의 설계 및 구현," 정보과학회논문지(C), Vol. 5, No. 2, pp.135-146, 1999

[2] 김용훈, "다양한 구조검색을 지원하는 XML 문서 검색기의 설계 및 구현", 석사 학위 논문, 충남대학교, 1999

[3] 유재수외 8명, "전자도서관 표준문서관리를 위한 XML 저장관리기 기술 개발", 한국지식웨어 최종보고서, 1999

[4] 박철현외 정재헌, 심대익, 이상구, "구조화된 문서에 대한 DBMS와 IRS의 성능 비교", '99 한국 데이타베이스 학술대회 논문집 15권 1호, pp218-225, 1999

[5] Charles L. A. Clarke, Gordon V. Cormack, Forbes J. Burkowski, "An Algebra for Structured Text Search and a Framework for its Implementation," The Computer Journal 38(1), pp. 43-56, 1995.

[6] Francois, "Generalized SGML repositories: Requirements and modelling," Computer Standards & Interfaces, pp.11-24, 1996

[7] Brian Lowe, Justin Zobel, Ron Sacks-Davis, "A Formal Model for Databases of Structured Text," DASFAA, pp.449-456, 1995

[8] Ian A. Macleod, "Storage and Retrieval of Structured Documents, Information Processing and Management," vol. 26, No., pp.197-208, 1990.

[9] Kyuchul Lee, Yongkyu Lee, Bruce Berra, "Management of Multi-structured Hypermedia Documents: A Data Model, Query Language, and Indexing Scheme," Multimedia Tools and Applications, pp.199-223, 1997

[10] Extensible Markup Language(XML) 1.0,"http://www.w3.org/TR/1998/REC-xml-19980210"

[11] Frakes, Baeza-yates, "Information Retrieval : Data Structure and Algorithms," Prentice-Hall, 1992.

[12] Marc Volz, Karl Aberer, Klemens Bohm, " An OODBMS-IRS Coupling for Structured Documents," Data Engineering., pp.34-42, 1996.

[13] Myaeng, S. H., Jang, D.-H., Kim, M.-S., and Zhoo, Z.-C., "A Flexible Model for Retrieval of SGML Documents," In Proc. of ACM SIGIR '98, pp.138-145, 1998

[14] Tuong Dao, "An Indexing Model for Structured Document to Support Queries on Content, Structure and Attributes," In Proc. of IEEE ADL '98, pp.88-97, 1998

강 형 일

1996년 목포대학교 전산통계학과(이학사). 1998년 목포대학교 전산통계학과(이학석사). 1998년 ~ 현재 충북대학교 정보통신공과학 박사과정 재학 중. 관심분야는 데이타베이스 시스템, XML, 정보검색

최 영 길

1985년 고려대학교 수학과 학사. 2000년 충북대학교 정보통신공학과 석사. 1987년 5월 ~ 현재 한국원자력연구소 선임 연구원. 관심분야는 GIS, 데이타마이닝, Axiomatic S/W Design

이 종 설

1996년 충북대학교 정보통신공학과(학사). 1999년 2월 ~ 현재 충북대학교 정보통신공학과 석사 과정. 관심분야는 데이타베이스 시스템, XML, 실시간 데이타베이스, 분산 객체 컴퓨팅.

유 재 수

1989년 전북대학교 공과대학 컴퓨터공학과(학사). 1991년 한국과학기술원 전산학과(공학석사). 1995년 한국과학기술원 전산학과(공학박사). 1995년 ~ 1996년 목포대학교 전산통계학과 전임강사. 1996년 ~ 현재 충북대학교 공과대학 전기전자공학부 조교수. 관심분야는 데이타베이스 시스템, 정보검색, 멀티미디어 데이타베이스, 분산 객체 컴퓨팅

조 기 형

1966년 인하대학교 전기공학과(공학사). 1984년 청주대학교 산업공학과(공학석사). 1992년 경희대학교 전자공학과(공학박사). 1988년 현재 충북대학교 공과대학 전기전자공학부 교수. 관심분야는 데이타베이스, 화상처리 및 통신, GIS, 통신프로토콜