韓國 컴퓨터情報學會 論文誌
第9卷 第4號, 2004. 12.

2004-9-4-1-5

효율적인 검색을 위한 Tree 형태의 XML 문서 구조 모델

김 영 란*

XML Structured Model of Tree-type for Efficient Retrieval

Kim, Young-Ran*

요 약

XML 문서가 DTD를 포함하지 않거나 여러 곳에서 XML 문서를 모았을 때, 그 구조는 비정규적일 수 있다. 비정규적인 구조를 가지는 문서들에 대해 정확한 구조적 질의를 작성하는 것은 어려운 일이다. 이 논문에서는 XML 문서의 효율적인 관리와 검색을 위한 XML 문서 모델과 구조적 검색 방법을 제안한다. 이를 위해 XML 문서의 구조 정보를 표현하기 위해 엘리먼트에 대한 정보를 갖는 고정된 크기의 LETID를 사용하고, 구조 정보 검색을 위한 부모 및 자식 엘리먼트 검색 알고리즘을 제시하였다. 제안된 방법을 이용하여 XML 문서의 구조 정보를 효율적으로 표현할 수 있을 뿐만 아니라, 간단한 연산으로 특정 엘리먼트에 직접적인 접근과 다양한 질의 처리가 가능하다. 또한, 특정 엘리먼트의 부모, 자식, 형제에 대한 다양한 구조 검색을 효율적으로 지원할 수 있는 효과가 기대된다.

Abstract

A XML Document has a structure which may be irregular. The irregular document structure is difficult for users to know exactly. In this paper, we propose the XML document model and the structure retrieval method for efficient management and structure retrieval of XML documents. So we use fixed-sized LETID having the information of element, describe the structured information retrieval algorithm for parent and child element to represent the structured information of XML documents. Using this method, we represent the structured information of XML document efficiently. We can directly access to specific element by simple operation, and process various queries. We expect the method to support various structured retrieval of specific element such as parent, child, and sibling elements.

▸ Keyword : XML Document Structure, Retrieval Model, Indexing Mechanism

• 제1저자 : 김영란
• 접수일 : 2004.10.14, 심사완료일 : 2004.11.13
* 충청대학 인터넷정보학부 부교수

Ⅰ. 서 론

e-business의 핵심은 인터넷을 매개체로 하는 분산 네트워크 컴퓨팅(distributed network computing)이라고 할 수 있다. 즉, 주어진 고유한 역할에 따라 상호 독립적으로 기능을 수행하는 복수의 서버 프로그램들이 네트워크를 통해 정보를 주고받음으로써 비즈니스 로직을 구현하게 되는 것이다. 이러한 비즈니스 로직을 구현하는 주요 프로그래밍 언어로 각광을 받고 있는 것이 Java 언어이고, 정보의 표현 방안으로 주목을 받고 있는 것이 XML (eXtensible Markup Language)이다. XML은 B2B 전자상거래 이외에도 컨텐트 제공자들 간의 컨텐트 관리, 기업 내부에 존재하는 다양한 형태의 이종 애플리케이션 간의 연동과 백오피스와의 연동, 데이타 웨어하우징 등에서 주로 이용되고 있다.

그동안 기업간 거래를 위해 사용되어 왔던 기술은 EDI(Electronic Data Interchange)이다. 이것은 서로 다른 기업 간의 상행위를 위한 전자문서를 통신회선을 매개체로 표준적인 규약(ANSI X.12, EDIFACT)을 이용해서 컴퓨터 간에 교환하는 행위로, VAN 사업자를 기반으로 이루어졌다. 이를 인터넷 기반에 적용시킬 경우 기존 EDI를 가장 잘 표현할 수 있는 방법이 XML인 것이다. 기존의 EDI는 Translator라고 하는 변환 모듈 매개체를 통해 문서를 번역하였고, 또한 프로그램을 통해 사람이 이해할 수 있도록 표현하였다. 그러나, XML에서는 이러한 중간 매개체가 없이도 문서의 데이타를 직접 사람이 이해할 수도 있으며, 데이타로서만 제공하는 것이 아니라 정보까지 같이 포함되어 있으므로 사용자들은 별도의 프로그램을 구현할 필요 없이 단지 웹 브라우저만으로 인식할 수 있다.

XML 문서는 기존의 문서와 달리 하나의 문서에 내용 정보와 구조 정보를 함께 가지고 있다. 따라서, 기존의 문서에서 제공하던 키워드를 기반으로 한 내용 정보에 대한 검색뿐만 아니라 문서의 논리적인 구조 정보에 대한 검색 기능이 필요하다. 이를 위해 XML 문서의 구조 정보를 표현하기 위한 여러 가지 방법들이 제안되고 있다[1][2]. 그러나 일부 방법들은 조상, 자손, 형제 관계의 엘리먼트에 접근하기 위해 복잡한 연산을 수행하거나 형제관계 노드를 알 수 없다. 또한 구조 트리의 깊이가 깊어질수록 엘리먼트 정보를 표현하기 위해 무한대의 저장 공간이 필요하다.

이 논문에서는 XML 문서의 구조 정보를 효율적으로 검색하기 위해 트리 구조를 이용한 XML 문서 모델을 제시하고, 이를 기반으로 한 구조 정보 검색 알고리즘을 정의하였다. 임의의 엘리먼트로부터 간단한 검색 과정을 통해 특정 엘리먼트에 쉽게 접근할 수 있도록 하기 위해, 고정된 크기의 노드 정보를 통해 엘리먼트 간의 계층 정보뿐만 아니라, 형제 노드의 순서를 표현하였고 검색의 질의와 문서 구조 검색의 유사도를 측정하는 방법을 고려하였다.

Ⅱ. 관련연구

사용자의 질의와 문서들간의 관련성 정도에 대한 순위(ranking) 결정을 위한 전통적인 검색 모델로는 크게 불리안 모델, 백터 모델, 확률 모델로 분류된다[3]. 불리언 모델은 집합을 기본 가정으로 하고 있으며 검색 결과는 질의 연산자들을 모두 만족하는 문서들의 집합이다. 백터모델은 문서 집합의 색인 개수에 대해, 문서의 색인어의 존재유무 및 가중치에 따라 매핑시킨다. 확률모델은 사용자의 질의가 어떤 특정 문헌에 관련 있을 확률과 관련 없을 확률을 구한 후, 두 확률의 비율에 따라 순위화를 결정한다.

XML로 표현된 구조화된 문서를 검색하기 위해서는 키워드에 의한 문서 단위의 내용 검색뿐만 아니라 엘리먼트를 기본 단위로 하는 구조 검색 및 애트리뷰트 검색이 지원되어야 한다[4][5][6]. 이를 위해 구조 정보를 효율적으로 표현하기 위한 연구가 선행되어야 한다. 현재 구조 정보를 표현하기 위한 모델로는 Subtree 모델, SCL 모델, K-ary 완전 트리 모델, 그리고 ETID 모델 등이 있다. Subtree 모델은 검색 효율을 향상시키기 위한 5가지 질의 명세를 제시하였다. 이들 질의 분류에 대한 검색을 지원하기 위한 인덱스 구조로서 XML 문서를 subtree 형태로 표현한 후 여기에 나타나는 모든 단위에 대해서 중복 색인을 하고 색인어가 위치하는 단위를 기록하는 방법을 제안하였다. 그러나 이 방법은 추출된 색인어가 나타난 계층의 모든 상위 계층에 대해서도 색인을 하므로 공간상의 중복이 일어난다는 단점이 있다. SCL 모델은 구조 문서의 계층적 관계보다는 포함 관계를 이용한 표현 방법으로서 SC-list (Simple

Concordance list)라는 데이터 타입을 통해 중첩된 정보를 허용하므로 리스트의 리스트와 같은 순환 구조를 다룰 수 있다는 장점이 있다. 이 모델은 텍스트와 마크업에 대해 색인 넘버를 부여한 후, 불용어를 제외한 텍스트 어휘들을 텍스트 인덱스에 색인 넘버로 저장하고, 마크업은 시작 태그와 종료 태그의 쌍으로 마크업 인덱스에 저장한다. 그러나 SCL 구조는 트리의 깊이를 표현할 수 없으므로 조상이나 형제 엘리먼트를 검색할 수 없다는 단점이 있다[7]. K-ary 완전 트리 모델은 문서에 대한 트리로부터 이들 노드 중 가장 큰 차수 K를 구하여 K-ary 완전 트리로 재구성한 후, 여기에 문서 트리를 매핑하여 각 노드에 노든 번호를 부여한다. 이 모델은 문서 구조 사이의 계층 관계를 간단한 공식을 통해 쉽게 구할 수 있다는 장점이 있는 반면, 매핑 과정에서 Null 노드가 많아질 수 있고 노드의 깊이가 깊어질수록 노드 변화가 커진다는 단점이 있다[8]. ETID 모델은 엘리먼트들 간의 계층 정보와 동일 부모 엘리먼트를 갖는 자식 엘리먼트들의 순서 정보, 그리고 동일한 부모 엘리먼트를 갖는 자식들 중 동일한 타입의 엘리먼트들에 대한 순서 정보를 통해 구조 문서를 표현한다. 이 방법은 기준 엘리먼트로부터 특정 엘리먼트에 대한 계층 정보와 순서 정보를 간단한 문자열 조작만으로 쉽게 구할 수 있다는 장점이 있는데 반해 트리의 깊이가 깊어질수록 각 노드를 표현하기 위한 공간이 무한대로 늘어난다는 단점이 있다[9].

III. XML 구조 정보에 대한 검색 모델

3.1 XML 문서 모델

XML 문서는 구조가 없는 단어들의 나열, 구조가 있는 트리(Tree) 또는 그래프(Graph)로 모델링한다. 구조가 없는 단어들의 나열로 모델링하는 것은 기존의 검색 모델을 XML 문서에 대해 그대로 적용하였을 경우이다. 트리로 모델링 하는 경우는 문서내의 링크 구조를 무시하고 엘리먼트 간의 포함관계를 트리로 그대로 나타낸 경우이다. 그래프는 트리로 나타내어진 구조상에서 링크구조를 분석하여 모델에 추가하게 됨으로써 모델링 될 수 있다[10].

XML 문서를 트리로 모델링 할 경우 그래프보다 쉽게 인덱스를 만들 수 있고 질의를 처리할 수 있다. 반면에 XML 문서가 가지고 있는 정보를 모두 모델링하지 못하게 된다. 이를 해결하기 위해 질의 언어가 트리로 모델링 된 문서의 링크 정보를 분석하여 처리한다면 XML 문서의 모든 정보를 활용 가능하게 될 수 있다. (그림 1)은 XML 문서에 대해, 트리 구조를이용하여 문서의 구조를 모델화 한 것이다.

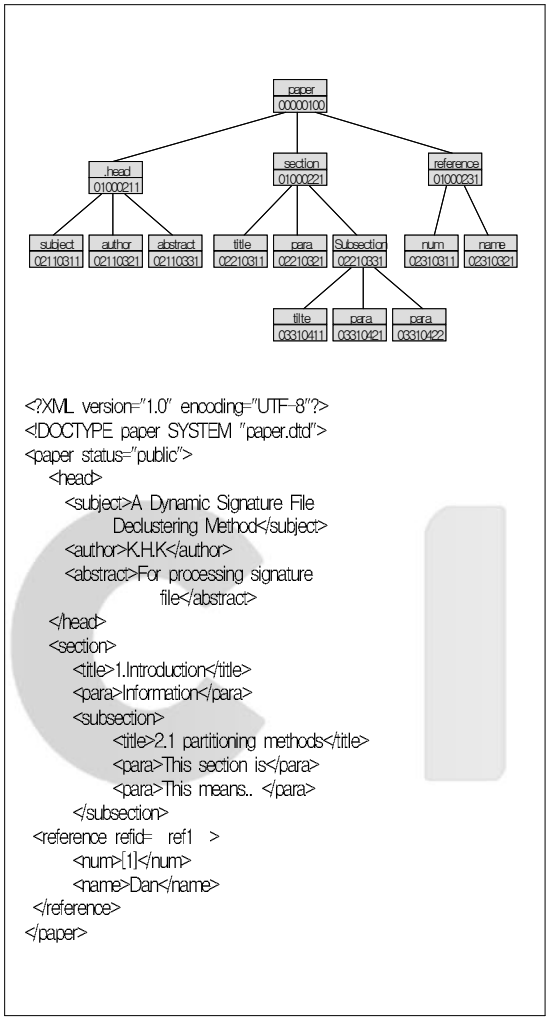


그림 1 트리구조를 이용한 XML 문서의 구조
Fig. 1 XML document information representation using Tree structure

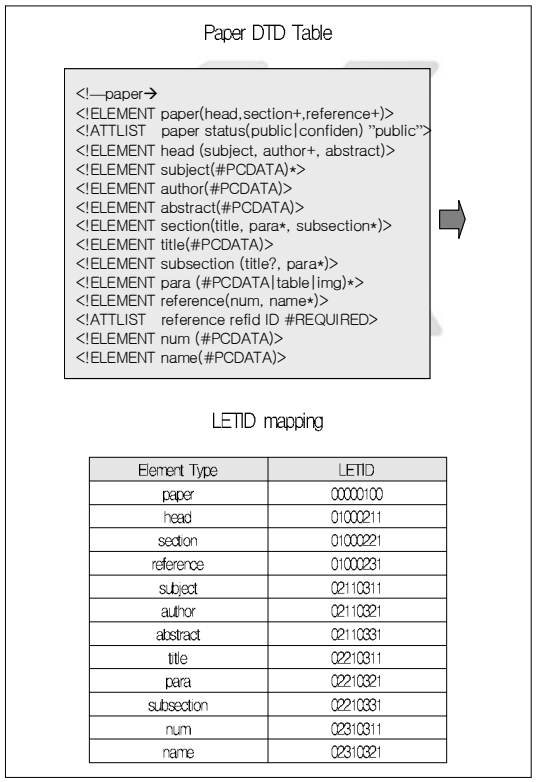
3.2 구조적 정보 표현 방법

XML 문서의 구조 정보를 검색하기 위해, 먼저 구조 정보 추출기를 통해 DTD로부터 엘리먼트 타입을 추출하고 엘리먼트 이름과 엘리먼트 ID를 연결시켜주는 LETID (Leveled Element Type ID) Mapping Table을 구성한

다. 인덱스 관리자는 LETID 정보를 통해 키워드, 엘리먼트, 그리고 애트리뷰트에 대한 색인 테이블을 구성하고, 저장 관리자를 통해 XML 문서 인스턴스를 저장한다.

XML 문서는 내용 정보 뿐만 아니라 구조 정보를 함께 표현한다. 따라서 XML 문서의 부모 자식 관계뿐만 아니라 해당 엘리먼트에 대한 형제 관계, 그리고 동일한 타입의 엘리먼트에 대한 순서 정보를 표현하기 위해 LETID를 이용한 구조 정보 표현을 제안한다. 이 방법을 통해 사용자는 간단한 연산으로 질의에 대한 결과를 쉽게 검색할 수 있으며 고정된 크기로 LETID를 표현하기 때문에 문서에 대한 계층 구조의 깊이에 상관없이 엘리먼트 정보를 표현하기 위해 추가적인 공간이 필요 없다.

LETID는 DTD에서 정의된 각 엘리먼트에 대한 고유값을 나타낸다. 엘리먼트에 고유 ID를 부여하는 방식으로 DTD의 논리적 구조를 분석할 때 부모, 형제 노드를 직접적으로 찾을 수 있고 ID 값에 깊이 정보가 포함되어 있기 때문에 고유 번호만으로 깊이를 알 수 있다.



Element Type	LETID
paper	00000100
head	01000211
section	01000221
reference	01000231
subject	02110311
author	02110321
abstract	02110331
title	02210311
para	02210321
subsection	02210331
num	02310311
name	02310321

그림 2 LETID 부여 방법
Fig. 2 The assigning method of LETID

LETID는 모든 엘리먼트의 노드를 고정된 8바이트로 표현하는데 각 바이트는 ASCII code 순서를 따르는 '0'→'9'→'A'→'Z'→'a'→'z' 순으로 구성된다. (그림 2)는 DTD에서 정의된 각각의 엘리먼트에 대해 LETID를 부여하고 이를 LETID 매핑 테이블로 표현한 것이다.

각각 2바이트로 엘리먼트의 부모, 자식간의 계층 정보와 형제간의 순서 정보를 나타낸다. 상위 4바이트(1234)는 부모 노드의 계층 정보와 부모 노드의 형제 노드에 대한 순서 정보를 유지한다. 이중 하위 2바이트(34)는 형제 엘리먼트들의 순서 정보와 동일한 타입의 형제 엘리먼트들 간의 순서 정보를 나타낸다. 마찬가지로 뒤의 4바이트(5678)는 현재 노드의 계층 정보와 현재 노드의 형제 노드에 대한 순서 정보를 표현한다. 각 엘리먼트들에 유일한 LETID 값을 부여하기 때문에 구조적 검색을 지원할 수 있고 반복적인 엘리먼트의 경우 서로 다른 LETID값을 부여하므로 다양한 구조 검색이 가능하다.

VI. 구조 정보 검색

4.1 알고리즘

구조 정보를 검색하기 위해서 제안된 색인으로부터 DID와 LETID 값을 구한다. 제안한 색인 모델은 부모 엘리먼트의 정보를 자식 엘리먼트에서 가지고 있으므로 구조 검색이 용이하며, 엘리먼트 정보를 표현한 LETID 값은 형제 엘리먼트들 간의 순서 정보까지 포함하고 있으므로 임의의 엘리먼트로부터 특정 엘리먼트를 검색하기 용이하다. 따라서 구조 정보 검색 알고리즘에서는 LETID값의 각 자리에 해당되는 정보를 이용하여 해당 엘리먼트를 구할 수 있다. 다음은 제안한 색인 구조를 이용한 구조 검색 알고리즘이다.

사용자 인터페이스로부터 전달받은 해당 엘리먼트 정보를 LETID 매핑 테이블로부터 구한다. 이때, LETID 정보와 N 단계 상위 부모를 나타내는 레벨 정보를 키 값으로 하여 색인 테이블로부터 해당 부모 노드를 검색한다. 즉, 전달받은 LETID 값의 상위 4자리 값을 하위 4자리 값으로 갖는 엘리먼트를 검색한다. 이때, N 단계 상위 부모를 검색하기 위해서는 전달받은 레벨 정보만큼 위의 단계를 반복하여 해당 엘리먼트를 구할 수 있다. (그림 3)은 특정 엘리먼

트의 부모 엘리먼트를 검색하는 알고리즘이다.

```
search_parent_element(letid, level) {
    curren_first_letid = letid의 앞 네자리 값;
    if (level = 0)          {return letid}
    else {
        parent_etid = get_element(element_table[i]);
        parent_level = parent_letid의 앞 두자리;
        while(level < parent_level) {
            parent_last_letid = parent_letid의
                    뒤 4자리 값;
            parent_level = parent_letid의 앞 두자리;
            if (current_first_letid = parent_last_letid) {
                    level--;
                    search_parent_element(parent_letid,
                    level);
            }
        parent_letid = get_element(element_table[i]);
        }
    }
}
```

그림 3 부모 엘리먼트 검색 알고리즘
Fig. 3 Retrieval algorithm for parent element

질의 분석기를 통해 사용자 인터페이스로부터 전달받은 엘리먼트 정보와 자식 엘리먼트의 순서 정보를 키 값으로 하여 색인 구조로부터 해당 엘리먼트를 검색한다. 이때, 자식 엘리먼트의 순서 정보는 LETID의 7번째 자리값으로 알 수 있다. 즉, 전달받은 엘리먼트의 LETID 값으로부터 하위 4자리 값을 상위 4자리 값으로 갖는 엘리먼트를 구한 다음, 이 엘리먼트의 LETID 정보로부터 7번째 자리 값과 전달받은 순서 정보를 비교하여 자식 엘리먼트를 검색한다.

```
search_child_element(letid, order) {
    curren_last_letid = letid의 뒤 네자리 값;
    child_letid = get_element(element_table[i]);
                //해당 엘리먼트의 자식 엘리먼트들을
                 가져온다.
    child_first_letid = child_letid의 앞 4자리 값;
    while(letid의 수) {
        if(current_last_letid=child_first_letid &&
          order=child_order_letid) return child_letid;
        else {
              child_letid = get_element(element_table[i]);
              child_first_letid = child_letid의 앞4자리 값;
        }
    }
}
```

그림 4 자식 엘리먼트 검색 알고리즘
Fig. 4 Retrieval algorithm for child element

4.2 실험 및 평가

구조 정보 표현 방법은 DTD에 나타나는 각 엘리먼트들에 대해 LETID를 부여하여 고유값을 주었다. LETID는 고정된 크기로 구성이 되며 각 자리에 부모의 계층 정보와 형제 노드의 순서 정보를 알 수 있도록 번호를 부여하였다. 제안한 색인 구조는 내용 검색을 지원하는 내용 색인, 구조 검색을 지원하는 구조 색인, 애트리뷰트 검색을 지원하는 애트리뷰트 색인으로 구성된다. 또한 제안한 색인 구조를 기반으로 한 구조 검색 알고리즘은 바로 위의 부모 엘리먼트 뿐만 아니라 N 단계 상위의 부모 엘리먼트와 N 번째 자식 엘리먼트를 구할 수 있다.

제안한 색인 모델의 비교 평가를 위해 기존 K-ary 완전 트리 구조 방법과 ETID에 의한 색인 모델을 비교 대상으로 하였다. 비교 방법으로는 구조 정보 표현 방법에 따른 각 엘리먼트 정보 의 디스크 검색 회수로 하였다. K-ary 방식은 기존 트리를 완전 트리로 변환하여 노드 값을 구하는 구조 정보 표현 방식을 적용함으로써, 디스크 검색 회수는 In+Rn*Pn+Sn*In(In:각 색인 파일 접근 회수, Rn : 검색결과, Pn: 포스팅 엔트리 크기, Sn: 구조색인 접근 회수)이 된다. 이는 노드의 깊이에 따른 노드 변화가 크고, 사용하지 않는 노드가 많아지므로 데이터 양이 커지는 단점이 있다. 반면에, 제안된 방식은 고정 크기의 LETID(Leveled Element Type ID)만으로 엘리먼트 정보를 표현함으로써, 디스크 검색 회수는 In+Rn*Pn이 된다. 이는 부모노드와 자식노드, 그리고 동일한 타입의 형제 엘리먼트의 순서 정보를 구할 수 있는 장점을 가지게 된다.

K-ary 완전 트리 구조 방법은 K값을 구하기 위해 각 색인 파일마다 구조 색인에 접근한다. 또한 문서 파싱을 통해 구한 형제나 자식 엘리먼트가 실제 문서에 존재하는 노드인지 가상 노드인지를 판별하기 위해 검색 결과 수만큼 다시 구조 색인에 접근하여야 한다. 그러나 제안한 색인 모델의 경우 LETID로 부모, 자식, 형제 노드의 정보를 모두 표현하므로 가상 노드 여부를 판별하기 위한 접근 시간이 줄어든다.

V. 결 론

XML 문서의 가장 큰 특징 중의 하나는 구조 검색이 가능하다는 점이다. 기존의 웹 문서는 주로 키워드를 기반으로 한 내용 검색을 통해 전체 문서를 검색하였다. 그러나 XML 문서는 문서의 구조 정보를 이용하여 특정 영역에 대한 검색이 가능하다. 이를 위해 XML 문서에 대한 구조 정보를 효율적으로 표현하고 이를 이용하여 구조 검색이 가능하도록 하여야 한다. 이 논문에서는 XML 문서의 구조 정보를 검색하기 위해 구조 정보를 표현하는 방법과 구조 검색을 지원하기 위해 색인 구조 및 검색 알고리즘을 제안하였다.

이와 같은 구조정보 표현과 색인을 이용하여 특정 엘리먼트에 직접적인 접근이 가능하고, 구조화 된 문서를 효율적으로 관리할 수 있으며, 다양한 질의처리가 가능하다.

향후 연구과제로는 제안한 방법을 기반으로 한 구조 문서 검색 시스템의 구현과 문서의 갱신 시 색인 모델을 적용하기 위한 연구가 필요하다.

참고문헌

[1] Brian Lowe, Justin Zobel, Ron Sacks-Davis "A Formal Model for Databases of Structured Text", Proceedings of the Fourth International Conference on Database Systems for Advanced Applications (DASFAA '95), pp449-456,1995

[2] Toung Dao " An Indexing Model for Structured Documents to Support Queries on Content, Structure and Attributes", Proceedings of ADL'98, pp,88-97, 1998.

[3] Ricardo A., Baeza-Yates and Gonzalo Navarro, Integrating Contents and Structure in Text Retrieval, SIGMOD Record, 1996.

[4] V. Christophides. et al, "From Structured Documents to Novel Query Facilities," ACM SIGMOD, pp. 313-324, Minesota, USA, 1994.

[5] Takeyuki Shimura, Masatoshi Yoshikawa, and Shunsuke Uemura Storage and Retrieval of XML Documents using Obkect-Relational Database, DEXA, pp. 206-217, 1999.

[6] 김영란 "XML DTD의 효율적인 검색을 구조 정보 및 인데스 메카니즘", 한국컴퓨터정보학회논문지, Vol. 8, No. 3, pp. 80-86, 2003.

[7] T. Dao, R.Sacks-Davis and J.A.Thom "An indexing scheme for structured documents and its implementation", In Proceedings of the 5th International Conference on Database Systems for Advanced Applications, pp 125-134, Melbourne, Australia, Aptial 1997.

[8] Sung-Geun Han, Jeong-Han Son, Jae-Woo Chang Zong-Cheol Zhoo "Design and Implementation of a Structured Information Tetrieval System for SGML Documents", IEEE, pp. 81-88, 1999.

[9] 박종관, 강형일, 손충범, 유재수 "XML 문서에 대한 효율적인 구조 기반 검색을 위한 색인 모델", '2000 추계 학술발표논문집, 한국정보과학회, pp, 18-20, 2000.

[10] 조윤기, 김영란 "스키마 기반의 XML 문서 관리 시스템 설계", 한국OA학회논문지, Vol. 6, No. 4, pp. 85-93, 2001.

저 자 소 개

김 영 란

충북대학교 대학원 전자계산학과 (이학박사)

현재 충청대학 인터넷정보학부 부교수

<관심분야> XML 문서 구조화, 정보 검색 시스템, 객체 지향 분석 및 설계