

XML DTD의 효율적인 검색을 위한 구조 정보 및 인덱스 메카니즘

김 영 란*

An Index Mechanism and Structure Information for Efficient Retrieval of XML DTD

Young-Ran Kim *

요 약

XML은 웹 상에서 정보를 전달하고 저장하는데 있어 가장 주목을 받고 있는 언어이다. XML로 표현된 정보는 의미가 부여됨으로서 보다 정확하고 빠른 검색을 제공한다. 반면에, XML 문서는 논리적으로는 유사하지만 구조적 측면에서 서로 다른 표현방식으로 작성됨으로 인해, 이 기종 시스템 및 장치를 갖는 문서 처리 환경에서 문서를 교환하고 공유하는데 어려움이 있다. 이 논문에서는 객체 지향 클래스 다이어그램을 XML DTD로 변환하고, 동일 주제에 대해 서로 다른 구조로 작성된 XML DTD문서를 효율적으로 관리하기 위한 구조 정보 표현과 인덱스 메카니즘을 설계하였다. 제안된 방법을 이용함으로써 특정 엘리먼트에 대한 효율적이고 빠른 검색을 지원할 수 있으며, 간단한 연산으로 엘리먼트에 접근을 용이하게 할 수 있는 효과가 기대된다.

Abstract

XML is being watched with keen interest for the communication and saving of information. Information represented in XML provides more accuracy and a higher-speed of reference after the process of being implication. But, it is difficult that XML document is exchanged or shared in different area such as electronic commerce or digital library. Because, XML document is being different in syntax but similar in logic, with using structured difference analysis. In this thesis, we converted object-oriented class diagram to XML DTD and designed an index mechanism based on the structure information for the converted XML DTD. With our methods, we could effectively and fastly retrieve the specific element and respect to usefully access element by simple operations.

▶ Keywords : XML DTD, 구조정보, 인덱스 메카니즘

* 충청대학 컴퓨터학부 교수

* 본 연구는 충청대학 교내연구비 지원에 의해 수행되었음

정보 표현 방법을 제안하고, 효율적인 검색을 지원하는 인덱스 메카니즘에 대해 기술한다. 4장에서는 인덱스 메카니즘의 성능평가를 기술하고, 5장에서는 결론과 향후 연구방향을 제시한다.

I. 서론

구조화된 문서를 지원하기 위한 방안으로 W3C (World Wide Web Consortium)에서는 1998년 HTML과 SGML의 장점을 수용한 표준 마크업 언어인 XML(eXtensible Markup Language)을 권고안으로 채택하였다[1].

XML은 SGML과 HTML의 장점을 수용하고 한계를 극복한 새로운 인터넷 표준이라 할 수 있다. XML로 작성된 문서는 그 용도와 목적 그리고 작성자 의도에 따라 표현과 구조가 다양하다. 대부분의 XML 저장 관리 시스템들은 자신의 시스템에서 정의한 문서 정의 형식인 DTD(Data Type Definition)나 XML 스키마를 기반으로 작성된 문서만을 취급한다. 따라서, 동일한 주제의 문서라도 서로 다른 문서 처리 환경에서 작성되었을 때, 서로 다른 형식의 문서로 취급된다. 이 경우 동일 주제에 대한 문서들을 검색하기 위해 그 문서의 구조를 정의한 DTD나 XML 스키마에 따라 서로 다른 질의를 작성해야 하는 문제점이 있다[2].

또한, 다양한 분야에서 XML을 이용하여 대량의 전자문서를 구축, 관리하고 있으나, 각 분야에서 생성된 문서의 교환을 통한 효율적인 저장 관리의 필요성이 점차 증대되고 있다[3][4][5].

이 논문에서는 객체 지향 분석과 설계를 위한 대표적인 모델링언어인 객체 지향 클래스 다이어그램을 XML DTD로 변환하고, 동일 주제에 대해 서로 다른 구조로 작성된 XML 문서를 통합 관리하기 위한 구조정보 표현 방법과 인덱스 메카니즘을 설계하고자 한다. 제안된 인덱스 메카니즘은 문서 계층상의 모든 레벨에서 내용기반 질의와 애트리뷰트 질의와 같은 다양한 질의를 지원하기 위한 구조정보를 기반으로 한다. 이 연구를 통해 기대되는 효과로는 서로 다른 구조로 작성된 XML DTD 기반의 문서를 단일 구조로 통합 관리함으로써 이 기종 시스템 및 환경간의 XML 문서 상호 교환이 가능하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 XML 문서의 모델링에 관한 연구동향 및 구조정보를 표현하고 검색하는 방법에 대한 기존 연구들을 살펴본다. 3장에서는 소프트웨어 산출물을 가시화하고 문서화하는데 사용되는 객체 클래스 다이어그램에 대한 XML DTD 변환 정보 추출 및 구조

II. 관련 연구

현재 XML에 대한 연구가 국내외적으로 활발히 진행되고 있으며, 대표적인 연구 분야로는 XML문서의 모델링에 관한 연구이다. SGML/XML 문서를 모델링하는 연구로는 구조도, XOMT 다이어그램, UML 클래스 다이어그램을 이용하는 방법이 있다. 이 중에서 UML(Unified Modeling Language) 클래스 다이어그램은 소프트웨어 중심 시스템의 산출물을 가시화하고, 명세화하고, 구축하고, 문서화하는데 사용되는 객체 지향 모델링 언어이다[6][7].

UML은 기초 구성요소들을 가지고 모델을 구성하는데, 특히 시스템의 정적인 부분을 가시화하기 위해 클래스 다이어그램, 객체 다이어그램, 컴포넌트 다이어그램, 배치도(Deployment Diagram)를 사용한다. 이와 같은 구성요소 중에서 특히 클래스 다이어그램은 시스템 설계에 핵심적인 역할을 차지하고 있기 때문에 주로 다루어진다.

Lander[8]는 XML 문법에 맞는 사상규칙과 알고리즘을 재정의 하였으며, 링크 구조에 대해서는 사상규칙 뿐만 아니라 형식 모델과 클래스 다이어그램 모델링 함수도 제안하였다. 또한 최근에는 XML 스크립트가 등장하여 기존의 XSL을 이용하여 XML 문서를 생성하는 것보다 빠르게 XML 문서의 생성 및 변형이 가능하게 되었다[9].

XML은 SGML의 내용모델에서 제외(exclusion)와 포함(inclusion)을 지원하지 않으며, 연결자로서 '&'를 지원하지 않는다. XML DTD는 XML에서 가장 중요한 부분으로 엘리먼트(element), 애트리뷰트(attribute), 엔티티(entity) 등의 구성요소로 나눌 수 있다. 이와 같은 특성을 갖는 XML문서에 대한 검색은 크게 내용검색, 구조검색, 그리고 내용과 구조가 혼합된 혼합검색으로 나눌 수 있으며 부가적으로 애트리뷰트 검색이 있다.

XML로 표현된 구조화된 문서를 검색하기 위해서는 키워드에 의한 문서 단위의 내용 검색뿐만 아니라 엘리먼트를 기본 단위로 하는 구조 검색 및 애트리뷰트 검색이 지원되

어야 한다[10]. XML 문서의 특성에 기반 한 질의를 지원하기 위한 구조 정보 표현 모델로는 SCL(Simple Concordance List) 모델, K-ary 완전 트리 모델, 추상화에 기반 한 모델 등이 있다.

SCL 모델은 정의된 문서 계층과 정의된 마크업 스키마로부터 독립을 제공한다. 이 모델은 문서구조에 대한 질의를 지원하기 위해 계층적인 관계보다 오히려 포함관계를 사용하고, 데이터 타입에서 중첩된 정보를 허용하기 때문에 순환구조를 다룰 수 있다[11][12].

그러나 색인을 포함하는 엘리먼트에 대해서는 알 수 있으나, 엘리먼트들에 대해 트리내의 깊이를 표현할 수 없다는 단점을 가진다.

K-ary 완전 트리 모델은 문서에 대한 트리로부터 이들 노드 중 가장 큰 치수 K를 구하여 K-ary 완전 트리로 재구성 한 후, 문서 트리를 매핑하여 각 노드에 번호를 부여한다[13]. 이 모델은 계산에 의해서 논리적 포함 관계인 엘리먼트를 빠르게 찾을 수 있는 장점이 있는 반면에, 노드의 깊이가 깊어질수록 노드 변화가 커지고 사용하지 않는 노드 번호가 많아지므로 복잡한 연산이 요구되는 경우가 있다.

추상화에 기반 한 방법은 구조화된 문서의 문법적 구조를 부모와 자식의 관계를 나타낼 수 있는 트리로 표현하여 문서구조를 나타낸다[14]. 이 모델은 실제 문서들의 구조를 추상화한 색인 구조인 추상화에 기반 한 구조 검색 색인을 설계하는 것이고 추상화는 순수 내용과 애트리뷰트에 적용될 수 있다. 반면에 추상화 되지 않은 문서 구조의 검색을 위해서는 문서 전체를 읽어야하는 오버헤드가 발생하고, 추상화의 치수에 따라 지원할 수 있는 구조 검색의 정도가 다르고, 형제 엘리먼트들의 순서를 알 수 없는 문제점을 갖고 있다.

III. 문서 저장 구조 설계

객체 클래스 다이어그램을 순수한 C++ 클래스 형태나 스키마 코드로 변환한 후, XML DTD를 생성하는데 실제로 UML 클래스 다이어그램이 의미하고 있는 내용을 정확하게 파악하여 변환하지 못하는 문제점을 갖고 있는 실정이다. 여기에서는 객체 클래스 다이어그램의 특성을 정확하게 XML DTD로 변환한 후, 효율적인 검색을 지원하기 위한 구조 정보 표현 방법과 인덱스 메커니즘을 제안한다.

1. 객체 클래스 다이어그램에 대한 XML DTD 변환 정보 추출

객체 클래스 다이어그램은 객체들의 타입들을 표현하며 그 클래스들의 정적인 관계를 표현하고 있다. 또한, 최근 들어 UXF의 등장으로 클래스 다이어그램 자체를 이 기종간에 교환할 수 있게 되었다.

객체 클래스 다이어그램을 순수한 C++ 클래스 형태나 스키마 코드로 변환한 후, XML DTD를 생성하는데 실제로 UML 클래스 다이어그램이 의미하고 있는 내용을 정확하게 파악하여 변환하지 못하고 문제점을 가지고 있는 실정이다. 그림 1에서 보는 바와 같이, UML클래스 다이어그램은 각각의 클래스들 간의 계층성은 표현하고 있지만, 동일한 레벨의 클래스들 간의 순서와 그러한 클래스들 간의 관계성에 대해서는 표현하고 있지 않다.

동일 레벨의 UML 클래스들 간의 관계는 순서가 있는 관계, or 관계, 순서가 없는 관계로 구성되어 있으며 XML DTD에서 순서가 있는 관계는 ".", or 관계는 "|", 순서가 없는 관계는 "&"를 이용하여 표기하였다. 어떤 클래스에 대한 발생 지시자의 경우는 0번 이상, 1번 이상, 0번 혹은 1번으로 구성되어 있으며 XML DTD에서 0번 혹은 1번은 "*", 0번 이상 반복은 "+", 1번이상 반복은 "?"를 이용하여 표기하였다. XML DTD 변환기는 통합 클래스 코드를 XML DTD로 변환하는 작업을 수행해야 한다. 통합 클래스 코드로부터 XML DTD를 생성할 때 다음과 같은 제한 조건을 두었다.

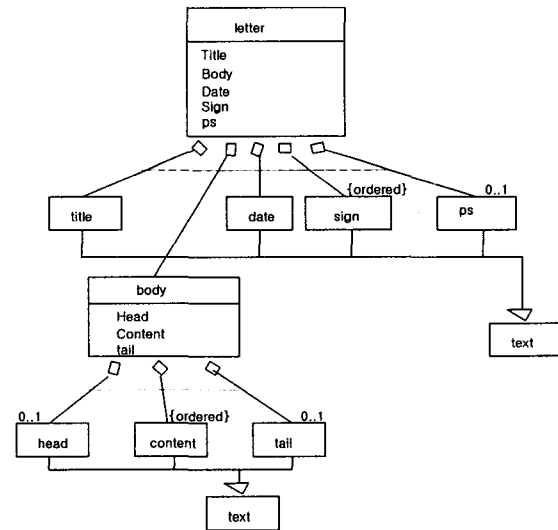


그림 1. 편지에 대한 객체 클래스 다이어그램
Figure 1. Object Class Diagram for letter

(제한조건 - 1) 모든 클래스는 반드시 'ELEMENT'로만 변환시킨다. 즉, Entity로의 변환은 생각하지 않는다.

(제한조건 - 2) "()"에 관한 처리는 생각하지 않는다.

(제한조건 - 3) "text"를 상속하는 클래스는 'ELEMENT'의 내용을 '(#PCDATA)'로만 간주한다.

위에서 제시된 연결자 처리 및 발생 지시자 처리 규칙을 기반으로 하여, <그림 1>의 UML 클래스 다이어그램에 대한 XML DTD는 다음과 같이 구성된다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE letter (
<!ELEMENT letter - o title,body,date,sign,ps?>
<!ELEMENT title - o (#PCDATA)>
<!ELEMENT body - o (head?,content,tail?)>
<!ELEMENT head - o (#PCDATA)>
<!ELEMENT content - o (#PCDATA)>
<!ELEMENT p - o (#PCDATA)>
<!ELEMENT tail - o (#PCDATA)>
<!ELEMENT date - o (#PCDATA)>
<!ELEMENT sign - o (#PCDATA)>
<!ELEMENT ps - o (#PCDATA)?)>
```

결과적으로 객체 클래스 다이어그램이 가지고 있는 객체 지향 개념을 객체 모델과 유사한 형태를 가지고 있는 XML DTD 형태로 변환함으로써 재사용성이 뛰어난 텍스트 형태의 문서로 저장하는 효과가 기대된다.

2. XML DTD에 대한 구조 정보 표현

XML 문서를 구성하는 기본 단위는 엘리먼트이다. 이 논문에서는 기존의 정보 검색 시스템에서 사용한 문서 단위를 위주로 한 검색이외에, 임의의 깊이에서 나타나는 엘리먼트 단위의 검색이 가능하도록 XML DTD에 대한 구조 정보 표현 방법을 제안하고자 한다. XML 문서에 대한 다양한 구조 검색을 처리하기 위한 표현 방법으로, XML 문서의 계층구조를 효율적으로 표현할 수 있는 순서화된 엘리먼트 타입 식별자를 이용하였다.

XML 문서는 내용 정보뿐만 아니라 구조 정보를 함께 표현하고 있다. 따라서, XML 문서의 부모 자식 관계뿐만 아니라 해당 엘리먼트에 대한 형제관계, 그리고 동일한 타입의 엘리먼트에 대한 순서 정보를 표현하기 위해 순서화된 엘리먼트 타입 식별자를 이용함으로써 간단한 연산으로 질의에 대한 결과를 검색할 수 있다.

각 계층의 엘리먼트에 대한 식별자는 두 바이트($X_1X_2, X_3X_4, \dots, X_nX_{n+1}$)로 표현하였으며, 각 바이트는 형제순서(sibling order)와 동일한 타입의 순서를 의미한다.

$X_1 X_2 X_3 X_4 \dots X_n X_{n+1}$	
X_1	: 첫 번째 계층에 있는 엘리먼트의 순서화된 엘리먼트 타입 식별자
X_2	: 첫 번째 계층에 있는 엘리먼트와 동일한 타입의 엘리먼트에 대한 순서정보
X_3	: 두 번째 계층에 있는 엘리먼트의 순서화된 엘리먼트 타입 식별자
X_4	: 두 번째 계층에 있는 엘리먼트와 동일한 타입의 엘리먼트에 대한 순서정보

엘리먼트 간의 계층정보에 대한 표현은 부모 엘리먼트의 정보를 유지함으로써 가능하게 된다. 앞에서 작성된 XML DTD에 대한 구조 정보를 표현하면, <그림 2>와 같다. 루트 엘리먼트의 순서화된 엘리먼트 타입 식별자를 '11'이라 할 때, <body>의 순서화된 엘리먼트 타입 식별자는 부모 엘리먼트의 순서화된 엘리먼트 타입 식별자에 <body>의 정보를 추가해 '1121'로 표현된다.

<body>의 순서화된 엘리먼트 타입 식별자 중, 뒤의 두 자리 '21'은 부모 엘리먼트에 대한 자식 엘리먼트의 순서정보와 동일한 부모의 자식 엘리먼트 중에서 동일한 이름을 갖는 엘리먼트 순서정보의 결합으로 표현된다. 즉, <body>는 <letter>의 두 번째 자식이며 <body>란 엘리먼트는 하나만 존재하므로 '21'로 표현된다. 또한, <content>는 <body>의 두 번째 자식이며 <content>란 엘리먼트는 하나만 존재하므로 '21'로 표현된다. 반면에 <content>의 자식 엘리먼트인 <sentence>의 경우, 동일한 이름의 3개 엘리먼트가 존재한다.

따라서, 자식 엘리먼트의 순서정보와 동일한 부모의 자식 엘리먼트 중에서 동일한 이름을 갖는 엘리먼트의 순서정보는 '11', '22', '33'이 부여된다.

자식 엘리먼트인 <sentence>는 이와 같은 결과물을 생성하기 위해 다음과 같은 과정을 수행한다.

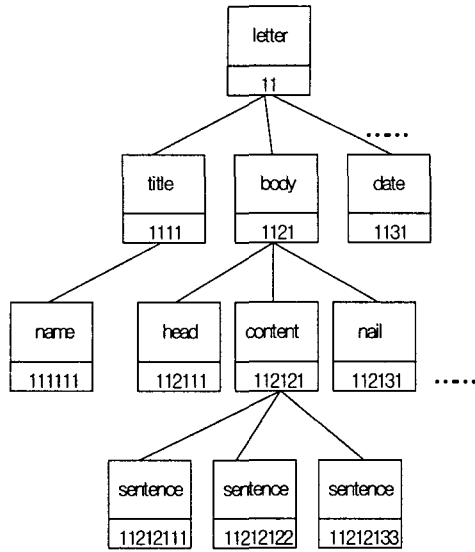


그림 2. XML DTD에 대한 구조 정보 표현
Figure 2. Represent structure information for XML DTD

전달받은 문서를 파싱하면서 시작 태그를 발견하면 이를 스택에 저장하고, 구조 정보 결과물에 추가한 후, 부모 엘리먼트에 대한 자식 엘리먼트의 순서정보 변수와 동일한 부모의 자식 엘리먼트 중에서 동일한 이름을 갖는 엘리먼트의 순서정보 변수의 값을 1씩 증가한다. 이렇게 구한 값은 현재 엘리먼트에 대한 고유의 식별 값을 구할 때 사용된다.

또한, 각 엘리먼트의 데이터 타입은 엘리먼트 데이터 타입 매핑 테이블을 참조하여 구한다. 만일 자식 엘리먼트가 있을 경우 현재 태그의 종료 태그를 추가한다. 현재 엘리먼트에 대한 종료 태그를 만나면 스택으로부터 시작태그를 삭제한다. 그 다음에 읽은 값이 부모 엘리먼트에 대한 종료태그일 경우에는 부모 엘리먼트에 대한 자식 엘리먼트의 순서정보 변수와 동일한 부모의 자식 엘리먼트 중에서 동일한 이름을 갖는 엘리먼트의 순서정보 변수의 값을 0으로 초기화한다.

3. 인덱스 메커니즘

XML DTD 문서의 효율적인 검색을 지원하는 인덱스 구성을 위해, 추출된 구조 정보를 기반으로 하여 XML DTD 문서에 기술된 엘리먼트의 구조 정보 매핑 테이블을 구성한다. 그림 2의 구조 정보에 대한 엘리먼트 이름과 엘리먼트 간의 계층정보를 나타내는 매핑 테이블을 구성하면 <표 1>과 같다.

Table 1. Mapping Table
표 1. 매핑 테이블

엘리먼트 타입	구조정보	엘리먼트타입	구조정보
letter	11	title	1111
body	1121	date	1131
name	111111	head	112111
content	112121	nail	112131
sentence	11212111	sentence	11212122

이와 같은 XML DTD로부터 추출한 구조 정보 결과물은 실제 문서의 검색을 위한 인덱스 파일을 생성한다. 이때 XML 문서에 대한 다양한 검색을 지원하기 위해 기존의 역파일 방식을 이용한 내용에 대한 인덱스, 구조에 대한 인덱스, 엘리먼트에 대한 인덱스를 포함한 인덱스 메커니즘은 <그림 3>과 같다. 내용에 대한 인덱스는 기존의 키워드 검색에서와 마찬가지로 문서 내에 있는 내용을 기반으로 한 검색으로 키워드에 대한 정보를 포함한 인덱스 파일과, 인덱스 파일이 참조할 포스팅 파일로 구성된다. 포스팅 파일은 키워드를 포함하고 있는 문서 정보와 문서내의 엘리먼트 정보로 구성된다. 따라서 내용에 대한 인덱스 파일을 통해 문서 단위의 검색뿐만 아니라 엘리먼트 단위의 검색이 가능하다.

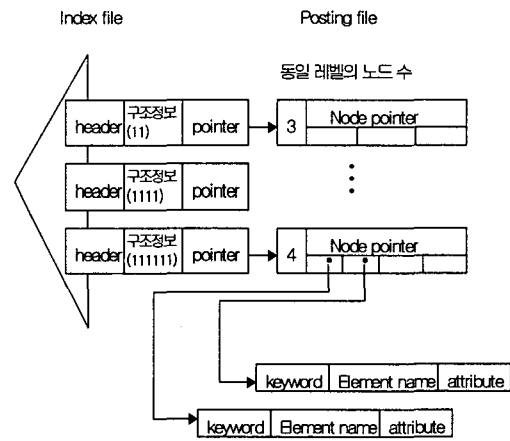


그림 3. 인덱스 메커니즘
Figure 3. Index Mechanism

엘리먼트에 대한 인덱스는 엘리먼트 기반의 구조 검색을 지원하기 위한 것으로 인덱스 파일은 엘리먼트 정보로 구성되며, 포스팅 파일은 키워드를 포함하고 있는 문서정보와

문서내의 엘리먼트 정보로 구성된다. 에트리뷰트에 대한 인덱스는 엘리먼트에 포함된 속성과 속성에 대한 값을 기반으로 한 검색으로 인덱스 파일은 에트리뷰트 이름과 값으로 구성되며, 이와 관련된 키워드를 포함하고 있는 문서정보와 문서내의 엘리먼트 정보로 포스팅 파일을 구성한다.

IV. 실험 및 평가

XML DTD 문서의 효율적인 저장을 위해 제안된 인덱스 메카니즘의 비교 평가를 위해 매핑 방식을 적용한 K-ary 완전 트리 구조 방법을 비교 대상으로 하였다.

K-ary 완전 트리 구조 방법은 문서에 대한 트리로부터 이들 노드 중 가장 큰 차수 K를 구하여 K-ary 완전 트리 구조를 재구성한 후, 문서 트리를 매핑하여 각 노드에 번호를 부여한다. 즉, i 번째 노드의 부모를 구하기 위해 연산식 $Parent(i) = (i-2) / K + 1$ 을 이용하고 i 번째 노드의 j 번째 자식을 구하기 위해 연산식 $Child(i, j) = K(i-1) + j + 1$ 을 이용하여 문서의 엘리먼트 간의 부모와 자식의 관계를 손쉽게 알 수 있는 장점이 있다. 이와 같이 문서 구조 사이의 계층 관계를 간단한 공식을 통해 쉽게 구할 수 있다. 반면에, 매핑 과정에서 Null 노드가 많아질 수 있고 노드의 깊이가 깊어질수록 노드 변화가 커진다는 단점이 있다. 즉, 문서 파싱을 통해 구한 형제나 자식 엘리먼트가 실제 문서에 존재하는 노드인지 가상 노드인지를 판별하기 위해 검색 결과 수 만큼 다시 구조정보에 대한 인덱스에 접근하여야 한다.

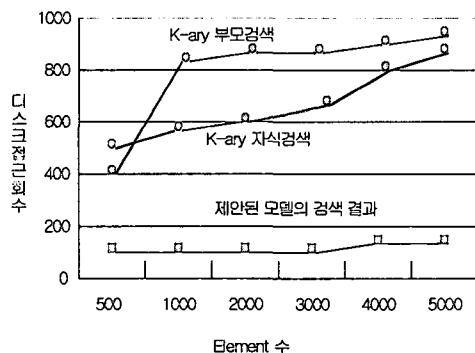


그림 4. 구조 검색 그래프
Figure 4. Structure retrieval graph

제안된 인덱스 메카니즘을 이용할 경우 추가되는 순서화된 엘리먼트 타입 식별자와 동일한 타입의 엘리먼트에 대한 순서정보를 위한 정보 없이, 하나의 정보로 부모, 자식, 형제 노드의 정보를 모두 표현하므로 가상 노드 여부를 판별하기 위한 접근 시간이 줄어든다. 부모와 자식에 대한 구조 검색을 수행할 때 엘리먼트 수의 증가에 따른 디스크 접근 회수의 변화를 측정된 결과는 그림 4와 같다. 이 방법에서는 부모 자식의 동일한 타입에 대한 형제 엘리먼트의 순서 정보를 매핑 구조로 부터 참조하여 인덱스에 추가시킴으로써 노드의 깊이에 따른 모드 변화가 작아지는 장점을 가지게 된다.

V. 결론

인터넷 상의 웹 문서, 전자 도서관, 인트라넷 상의 CALS(Continues 문 Acquisition Life Cycle Support), 그리고 전자상거래 등 다양한 분야에서 XML을 이용하여 대량의 전자문서를 구축, 관리하고 있으나, 각 분야에서 생성된 문서의 교환을 통한 효율적인 저장관리의 필요성이 점차 증대되고 있다.

이 논문에서는 객체 지향 클래스 다이어그램으로부터 변환된 XML DTD에 대해, 동일 주제에 대해 서로 다른 구조로 작성된 XML문서를 통합 관리하기 위한 문서 관리 시스템을 설계하였고 XML 문서의 구조 정보를 저장하는 방법과 효율적인 인덱스 메카니즘을 제안하였다. 제안된 인덱스 구조는 내용검색을 지원할 뿐만아니라 구조 검색을 지원하는 엘리먼트에 대한 인덱스와 에트리뷰트에 대한 인덱스로 구성하였다.

결론적으로 제안된 구조정보 표현과 인덱스 메카니즘을 이용하여 특정 엘리먼트에 직접 접근이 가능하고, 구조화된 문서를 효율적으로 관리할 수 있는 효과가 기대된다. 향후 연구과제로는 제안된 방법을 기반으로 한 자동화된 문서 검색 시스템을 구현하는 연구가 요구된다.

참고문헌

- [1] W3C, Extensible Markup Language(XML) Version 1.0, <http://www.w3.org/TR/REC-xml>, Feb., 10, 1998.
- [2] 나홍석, 채진석, 김창화, 백두권, "차세대 웹사에서의 문서교환 및 검색을 위한 프레임워크", 정보처리 제 6 권, 제 3호, pp. 52-61, 1999
- [3] Patricia Francois, Piere Bazex, "SGML/HyTime Repositories : Requirements and Data Modeling using Object-Oriented Database Concepts", DEXA, 1995.
- [4] Hiroshi Maruyama, Kent and Naohiko Uramoto, "XML and Java Developing Web applications", Addison Wesley Longman, 1999.
- [5] Hyung-Il Kang, Jae Soo Yoo, ByoungYup Lee, "Design and Implementation of a XML Repository System using DBMS and IRS", XML Asia Pacific, 2000.
- [6] Booch, Jacobson, Rumbaugh, "The Unified Modeling User Guide", Addison-Wesley, 1999.
- [7] Craig Larman, "Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design" Prentice-Hall, 1998.
- [8] Richard Lander, "ML : The New Markup Wave", http://www.esclub.ca.u/relander/XML/Wavw/xml_mw.html
- [9] E. Herwijnen, "Practical SGML", 2nd Edition, Kluwer Academic Publishers, 1994.
- [10] V. Christophides. et al, "From Structured Documents to Novel Query Facilities", ACM SIGMOD, pp. 313-324, 1994.
- [11] Tuong Dao, Ron Sacks-Davis, James A. Thom, "An Indexing Sacks-Davis Structures Documents and its Implementation", Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, pp. 125-134, 1997.
- [12] Tuong Dao, "An indexing Model for Structured Documents to Support Queries on Content, Structure and Attributes", Proceedings of ADL'98, 1998.
- [13] Sung-Geun Han, Jeong-Han Son, Jae-Woo Chang, "Design and Implementation of a Structured Information Retrieval System for SGML Documents", IEEE, pp. 81-88., 1999.
- [14] Chow, J. H., Cheng J., Chang D., "Indexing Design for Structured Documents based on Abstraction", Proceedings of the 6th International Conference on Database Systems for Advanced Applications, pp. 89-96, 1999.

저자소개

김영란

1988 충북대학교 전산통계학과 졸업(이학사)

1991 충북대학교 대학원 전자계산학과 졸업(이학석사)

1997 충북대학교 대학원 전자계산학과 졸업(이학박사)

현재 충청대학 컴퓨터학부 교수