SGML 문서의 효율적인 검색을 위한 색인 및 질의 언어의 설계 및 구현

이 봉 신[†] · 이 경 호[††] · 고 승 규[†] · 최 윤 철[†††]

요 약

본 논문에서는 SGML 문서의 효율적인 검색을 위한 방법론을 제안한다. 이를 위하여 메타 데이터, 색인 범위, 엘리먼트 이름의 재정의, 그리고 데이터베이스 생성에 관한 정보를 기술할 수 있는 언어로 IDDL을 정의한다. 또한 메타 데이터와 구조에 대한 복합 질의를 효과적으로 표현할 수 있는 질의 언어인 IDQL을 제안한다. 본 논문에서는 제안된 방법론의 유효성을 입증하기 위하여 IDDL과 IDQL에 기반한 검색시스템을 개발하였으며 이를 상이한 구조의 대용량의 문서 집합에 실험하였다. 그 결과, 제안된 방법론은 데이터베이스를 동적으로 구성하며 사용자에게 편리한 검색 환경을 제공하였다.

Design and Implementation of Indexing and Query Languages for an Efficient Retrieval of SGML Documents

Bong-Shin Lee[†] · Kyong-Ho Lee[††] · Seung-Kyu Ko[†] · Yoon-Chul Choy[†††]

ABSTRACT

We present new methods for an efficient retrieval of SGML documents. We define IDDL (index database description language) which is able to describe various information such as meta data, an indexing range, and the creation and manipulation of a database. In addition, we design IDQL (index database query language) that can deal with querying meta data as well as logical structure. Especially, the retrieval system based on IDDL and IDQL has been developed and implemented, and has been experimented on large number of documents. Experimental result shows that the proposed method provides the dynamic creation of an index database and a convenient retrieval environment.

1. 서 론

SGML(Standard Generalized Markup Language)[1, 2, 3, 4, 5, 6, 7, 8]은 이 기종간에 호환이 가능하고 논리적인 구조 정보를 포함한다는 장점 때문에 CALS(Commerce At Light Speed), EC(Electronic Commerce)/EDI(Electronic Data Interchange), 인트라넷/그룹웨어, 전자도서관 등의 다양한 분야에서 문서 처리의 표준 포맷으로 자리 잡았다. 이에 SGML 문서의 효율적인 관리를 위한 검색에 관한 연구가 활발히 진행 중이다.

기존의 전문 검색 시스템은 문서를 단어의 연속으로 간주하고 항상 문서 전체에 대하여 검색하기 때문에 검색 효율이 낮고 SGML 문서의 중요한 특징인 구조에 기반한 검색을 지원하지 않는다. 이에 SGML 문서의 구조에 기반한 검색에 대한 연구가 활발히 진행 중이다[9, 10, 11, 12, 13, 14 ,15, 16, 17, 18, 19, 20, 21]. 그

※ 본 논문은 한국과학재단 핵심전문연구과제(과제번호: 961-0902-012-2)의 연구비 지원에 의한 결과임.
† 준 회 원 : 연세대학교 대학원 컴퓨터과학과
†† 정 회 원 : 연세대학교 대학원 컴퓨터과학과
††† 정 회 원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 1999년 4월 26일, 심사완료 : 1999년 10월 12일

러나 대부분의 기존 연구는 DTD(Document Type Definition)[1]로부터 데이터베이스 스키마의 자동 생성 기법과 정교한 구조 검색 기법에 초점이 맞추어져 있다.

이에 데이터베이스 스키마가 DTD의 구조 및 엘리먼트(element)의 이름에 의존적이기 때문에 사용자는 검색을 위하여 해당 DTD의 구조 및 엘리먼트의 이름을 사전에 숙지하여야 한다. 또한 유사한 구조의 문서 집합에 대하여 동일한 검색 환경을 지원하지 못한다. 이러한 문제점을 해결하기 위해서는 저장 시 엘리먼트의 이름을 재 정의할 수 있는 방법이 요구된다. 일반적으로 방대한 양의 SGML 문서의 경우, 유연한 검색 환경을 지원하기 위하여 문서의 종류에 따라 검색 범위를 제한할 수 있는 방법이 요구된다. 한편 SGML 문서는 논리적인 구조 정보는 물론이고 저자나 작성일 등의 부가적인 정보를 포함한다. 본 논문에서는 이를 메타 데이터(meta data)라고 정의한다. 기존의 연구 결과는 메타 데이터 및 구조에 대한 복합적인 검색 환경을 지원하지 않는다.

이에 본 논문에서는 대용량의 SGML 문서 집합에 대한 효율적인 검색을 지원하기 위하여 실용적이며 유연한 방법론을 제안한다. 먼저 효율적인 색인어 데이터베이스의 생성을 위하여 메타 데이터, 색인 범위, 그리고 데이터베이스 스키마의 생성에 관한 정보 등을 기술할 수 있는 언어로 SGML의 응용인 IDDL(Index Database Description Language)을 제안한다. IDDL은 유사한 구조의 문서 집합에 대하여 동일한 인터페이스와 사용자의 편리성을 지원하기 위하여 질의에 사용되는 엘리먼트와 실제 엘리먼트의 이름간의 사상 함수(mapping function)에 기반 한다. 또한 IDDL에 의하여 생성된 데이터베이스에 대하여 메타 데이터와 구조 기반의 내용 검색을 복합적으로 표현할 수 있는 질의 언어인 IDQL(Index Database Query Language)을 설계하였다.

특히 본 논문에서는 제안된 방법론의 성능을 분석하기 위하여 IDDL과 IDQL에 기반한 검색 도구를 개발하고 이를 대용량의 문서 집합에 실험하였다. 실험 결과, 제안된 방법론은 IDDL 정보에 따라 데이터베이스를 동적으로 구성하기 때문에 상이한 구조를 갖는 대용량의 문서 집합에 대하여 문서 환경에 적합한 데이터베이스를 효과적으로 생성하였다. 또한 IDDL은 전술한 바와 같이 문서의 구조에 대한 사상 함수에 기반하기 때문에 사용자에게는 보다 편리한 검색 환경을 제공하며, SGML의 응용이므로 데이터베이스 관리자에게는 용이한 사용 환경을 제공하였다. 또한 메타 데이터와 구조에 기반한 전문 검색을 복합적으로 지원하였다.

본 논문의 구성은 다음과 같다. 2절에서는 IDDL에 대한 자세한 설명과 사용 예를 기술하고, IDQL의 특징을 소개한다. 3절에서는 제안된 방법론에 기반한 검색 도구의 설계 및 구현에 대하여 기술하고, 4절에서는 검색 도구의 주요기능과 성능 분석에 대한 고찰을 기술한다. 마지막으로 5절에서는 결론을 기술한다.

2. IDDL과 IDQL

SGML 문서의 검색과 관련한 기존 연구의 대부분은 DTD로부터 데이터베이스 스키마의 자동 생성과 구조 기반의 검색에 그 초점이 맞추어져 있다. 이에 전술한 바와 같이 다음과 같은 문제점을 갖고 있다. 첫번째는 메타 데이터를 나타내는 엘리먼트와 문서의 구조를 나타내는 엘리먼트를 구별할 수 없으며, 색인 및 검색의 대상이 되는 엘리먼트를 제한할 수 없다. 두 번째로 기존의 방법은 엘리먼트의 이름에 의존적이기 때문에 유사한 구조의 문서 집합에 대한 동일한 검색이 불가능하며 일반 사용자는 검색의 대상이 되는 해당 DTD의 구조 및 엘리먼트의 이름을 사전에 숙지하여야 한다.

예를 들어 (그림 1)의 DTD는 문서의 논리적인 구조 이외에 저자 및 작성일을 나타내는 엘리먼트인 *author*와 *date*를 포함한다. 한편 구조에 해당하지 않

```
<!ELEMENT  doc      - -  (info, body)                  >
<!ELEMENT  info     - -  (author, date)                -- 메타 데이터 - >
<!ELEMENT  author   - -  (#PCDATA)                     -- 저자-->
<!ELEMENT  date     - -  (#PCDATA)                     -- 작성일 -->
<!ELEMENT  body     - -  (title, abstract, content, ref) -- 전문 -->
<!ELEMENT  title    - -  (#PCDATA)                     -- 제목 -->
<!ELEMENT  abstract - -  (korean, english)             -- 요약 -->
<!ELEMENT  korean   - -  (para+)                       -- 국문요약 -->
<!ELEMENT  english  - -  (para+)                       -- 영문요약 -->
<!ELEMENT  content  - -  (intro, main, close)          -- 본문 -->
<!ELEMENT  intro    - -  (para+)                       -- 서론 -->
<!ELEMENT  main     - -  (para+)                       -- 본론 -->
<!ELEMENT  close    - -  (para+)                       -- 결론 -->
<!ELEMENT  ref      - -  (para+)                       -- 참고문헌>
<!ELEMENT  para     - -  (#PCDATA)                     >
```

(그림 1) SGML DTD의 예

으며 단순히 문자열을 포함하는 엘리먼트인 *para*를 포함한다. 이에 색인어 데이터베이스의 생성 시 이러한 사항을 반영하여야 한다. 한편 검색의 효율 및 사용자의 편리성을 높이기 위하여 엘리먼트의 이름을 재 정의할 필요가 있다. 예를 들어 엘리먼트 *intro*, *main*, 그리고 *close*를 서론, 본론, 그리고 결론으로 재 정의할 수 있다면 사용자에게 보다 편리한 검색 환경을 지원할 것이다.

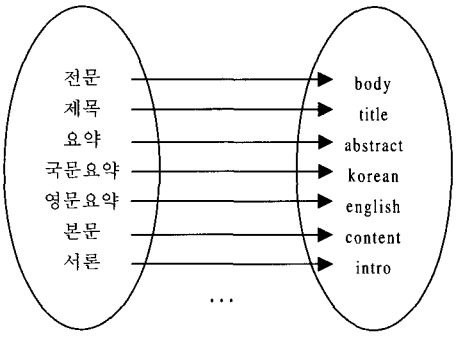
따라서 본 논문에서는 기존 연구의 문제점을 해결하고 보다 유연한 검색시스템을 지원하는 방법론을 제안한다. 먼저 메타 데이터, 색인의 대상이 되는 엘리먼트의 지정, 엘리먼트 이름의 재정의, 그리고 데이터베이스 스키마에 대한 정보를 기술할 수 있는 언어로 SGML의 응용인 IDDL을 제안한다. 특히 본 논문에서는 IDDL에 따라 기술된 SGML 문서를 IDDL 명세(specification)라고 정의한다. 따라서 데이터베이스 제공자가 기술한 IDDL 명세에 따라 동적으로 데이터베이스를 구성하기 때문에 검색 도구의 재활용이 가능하고 데이터베이스를 효과적으로 생성할 수 있다. 또한 IDDL은 엘리먼트 이름의 재정의를 지원하는 사상 함수에 기반하기 때문에 사용자에게 보다 직관적인 검색 인터페이스를 지원한다. 한편 메타 데이터에 대한 검색과 구조 기반의 내용 검색을 복합적으로 지원할 수 있는 질의 언어인 IDQL을 설계하였다. IDDL과 IDQL에 대한 보다 자세한 설명은 다음과 같다.

2.1 IDDL

SGML은 본래 문서의 논리적인 구조 정보를 기술하기 위하여 개발되었으나 이의 유연한 프로그래밍 언어적 특성 때문에 SGML은 다양한 응용 분야에 활용 가능하다. 이에 본 연구에서는 SGML의 응용으로 색인 및 검색에 필요한 정보를 기술할 수 있는 언어로 IDDL을 개발하였다. IDDL은 문서의 구조, 검색범위, 검색에 사용할 필드(field), 그리고 연산자 등에 대한 기술이 가능할 뿐만 아니라 테이블(table) 및 관계(relation)의 생성이나 레코드(record)의 추가 등의 데이터베이스의 조작에 관한 정보를 기술할 수 있다. 따라서 데이터베이스 생성자는 새로운 언어를 배울 필요 없이 SGML을 이용하여 색인어의 저장방법 및 검색방식을 정의할 수 있다. IDDL의 주요 부분에 대한 보다 자세한 설명과 사용 예는 다음과 같다. IDDL의 전체 구성은 부록에 첨부하였다.

2.1.1 사상 함수에 기반한 구조 기술

사용자에게 보다 친숙한 검색 환경을 제공하기 위해서는 문서의 구조를 사용자의 관점에서 질의문에 표현할 수 있어야 한다. 이를 위해 질의문에 사용되는 문서의 구조를 실제 문서의 구조로 사상하는 함수를 정의하였다. 사상 함수는 질의문에 사용되는 엘리먼트의 집합으로부터 실제 SGML 문서의 엘리먼트 집합으로의 일대일 함수이다. (그림 2)는 (그림 1)의 DTD에 사상 함수를 적용한 예이다.



(그림 2) 사상 함수의 적용

효과적인 검색을 위해서는 동일한 구조의 문서 집합뿐만 아니라 유사한 구조의 문서 집합에 대한 검색이 가능하여야 한다. 문서가 계층적인 트리 구조로 표현될 수 있는 것과 마찬가지로 문서의 종류 자체도 계층적인 트리 구조로 표현이 가능하다. 이에 문서의 종류 사이의 계층적인 포함관계와 색인의 대상이 되는 엘리먼트를 표현하기 위하여 (그림 3)과 같이 *DOCMODEL* 엘리먼트와 *NODE* 엘리먼트를 정의하였다.

```
<!ELEMENT    DOCMODEL  - - ( NODE+ )              >
<!ELEMENT    NODE      - - ( NODE* )              >
<!ATTLIST    NODE      NAME CDATA #REQUIRED
                       ISRANGE      (%bool;)  TRUE  >
```

(그림 3) DOCMODEL 엘리먼트와 NODE 엘리먼트

(그림 4)는 (그림 2)의 사상 함수를 바탕으로 *DOCMODEL* 엘리먼트와 *NODE* 엘리먼트를 이용하여 문서의 종류, 검색대상인 엘리먼트, 그리고 엘리먼

트 간의 포함 관계를 기술한 예이다. 특히 *NODE* 엘리먼트의 속성(attribute)인 *NAME*과 *ISRANGE*에 값을 명시함으로써 문서의 종류와 검색의 대상이 되는 엘리먼트를 지정한다. 예를 들어 논문지는 *ISRANGE*의 값이 *FALSE*이므로 문서의 종류를 의미하며 전문, 제목, 요약, 국문요약, 영문요약, 내용, 서론, 본론, 결론, 그리로 참고문헌 등의 엘리먼트는 검색의 대상이다. 한편 학회지라는 문서의 종류를 포함하고 싶으면, 학회지의 문서 구조에 해당하는 트리 구조를 생성하여 이를 논문지의 형제 노드(sibling node)로 추가하면 된다. 또한 논문지와 학회지를 정보처리학회에 속하는 문서의 종류로 만들려면 정보처리학회 노드를 논문지와 학회지의 부모 노드(parent node)로 삽입하면 된다.

```
<DOCMODEL>
    <NODE NAME=논문지 ISRANGE=FALSE>
        <NODE NAME=전문>
            <NODE NAME=제목></NODE>
            <NODE NAME=요약>
                <NODE NAME=영문요약></NODE>
                <NODE NAME=국문요약></NODE>
            </NODE>
            <NODE NAME=본문>
                <NODE NAME=서론></NODE>
                <NODE NAME=본론></NODE>
                <NODE NAME=결론></NODE>
            </NODE>
            <NODE NAME=참고문헌></NODE>
        </NODE>
    </NODE>
</DOCMODEL>
```

(그림 4) 문서 구조의 기술

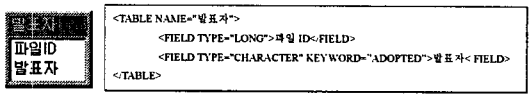
2.1.2 데이터베이스 스키마 및 메타 데이터 기술

색인어 데이터베이스를 생성하기 위해서는 테이블, 관계, 그리고 레코드에 대한 조작이 가능해야 한다. 한편 효과적인 문서 검색을 위하여 메타 데이터에 대한 지원은 필수적이다. 이에 문서의 구조와 더불어 메타 데이터에 대한 테이블 스키마를 기술하여야 한다. 테이블은 하나 이상의 필드로 이루어져 있으며 주 키(primary key) 필드를 가질 수 있다. 필드는 검색대상이 되는 것이 있으며 단지 정보를 저장하기 위한 것이 있다. 필드는 자료 유형, 크기, 널(null)의 여부 등에 대한 속성 이외에도 검색대상을 나타내는 속성이 있다. 테이블과 필드에 대한 정보를 기술하기 위한 *TABLE* 엘리먼트와 *FIELD* 엘리먼트에 정의는 (그림 5)와 같다.

```
<!ELEMENT TABLE - -          ( FIELD+ )                    >
<!ATTLIST  TABLE NAME        CDATA  #REQUIRED
                 PRIMARYKEY  CDATA  ""                     >
<!ELEMENT FIELD  - -         CDATA                         >
<!ATTLIST  FIELD TYPE        (%datatype;)   CHARACTER
                 SIZE        NUMBER         50
                 NOTNULL     (%bool;)       TRUE
                 KEYWORD     (%keywordtype;) IGNORED       >
```

(그림 5) TABLE 엘리먼트와 FIELD 엘리먼트

(그림 6)은 *TABLE*과 *FIELD* 엘리먼트를 이용하여 테이블을 정의한 예이다. 발표자 테이블은 정수형(long integer)의 자료 유형을 갖는 파일ID와 문자열 유형의 발표자 필드로 구성되며 특히 발표자 필드는 검색 대상이다.



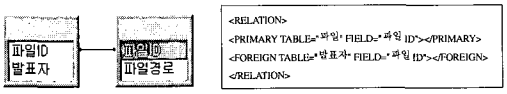
(그림 6) 테이블의 생성

한편 (그림 7)의 *RELATION* 엘리먼트를 이용하여 테이블 간의 관계 정보를 기술한다. 본 논문에서는 데이터베이스에 문서의 파일 경로를 저장하고 실제 문서는 파일 시스템을 이용하여 관리한다. 이에 검색시에 문서의 경로를 얻기 위해서 관련 테이블 간의 조인(join) 연산을 수행한다. 동적으로 구성된 데이터베이스에서 테이블간의 조인 정보를 얻어야 하기 때문에 테이블 간의 관계에 대한 정보가 필요하다. 테이블 간의 관계 정보를 이용하여 검색필드가 속한 테이블에서 파일경로가 저장되어 있는 테이블까지의 경로를 계산한다.

```
<!ELEMENT RELATION  --          (PRIMARY, FOREIGN)     >
<!ELEMENT PRIMARY   - O         EMPTY                  >
<!ELEMENT FOREIGN   - O         EMPTY                  >
<!ATTLIST PRIMARY   TABLE       CDATA    #REQUIRED
          FIELD     CDATA       #REQUIRED              >
<!ATTLIST FOREIGN   TABLE       CDATA    #REQUIRED
          FIELD     CDATA       #REQUIRED              >
```

(그림 7) RELATION 엘리먼트

(그림 8)은 *RELATION* 엘리먼트를 이용하여 두 테이블 사이의 관계를 기술한 예이다. 파일 테이블과 발표자 테이블 간에는 파일ID 필드를 통해서 관계가 형성 된다. 예를 들어 "발표자 = 홍길동" 인 질의가



(그림 8) RELATION 엘리먼트 기술 예

```
<!ELEMENT   RECORD          - -     (VALUE| SELECTEDVALUE)* >
<!ELEMENT   VALUE           - -     ( #PCDATA )             >
<!ELEMENT   SELECTEDVALUE   - -     ( SELECT )              >
<!ELEMENT   SELECT          - -     ( FROM, WHERE )         >
<!ELEMENT   FROM            - -     ( #PCDATA )             >
<!ELEMENT   WHERE           - -     ( #PCDATA )             >
<!ATTLIST   RECORD          TABLE   CDATA   #REQUIRED >
<!ATTLIST   VALUE           FIELD   CDATA   #REQUIRED >
<!ATTLIST   SELECTEDVALUE   FIELD   CDATA   #REQUIRED >
<!ATTLIST   SELECT          FIELD   CDATA   #REQUIRED >
```

(그림 9) 레코드의 생성 관련 엘리먼트

주어졌을 때 발표자 테이블과 파일 테이블 간의 관계 정보를 이용하여 파일경로를 계산할 수 있다. 테이블, 필드, 그리고 관계를 생성한 후에 메타 데이터를 저장하기 위하여 테이블의 필드에 고정된 값이나 이미 저장되어 있는 값을 찾아서 삽입한다. (그림 9)의 *RECORD* 엘리먼트는 이러한 정보를 기술한다.

(그림 10)은 *INSERT*와 *RECORD* 엘리먼트를 이용하여 데이터를 입력한 예이다. 출원 테이블의 경우 출원번호, 출원일자, 그리고 출원종류 필드는 문서로부터 직접 추출한 값을 지정하였으며 파일 필드에는 파일 테이블의 특정 파일경로에 해당하는 파일 ID를 참조하도록 지정하였다.

전술한 바와 같이 *DOCMODEL*과 *NODE* 엘리먼트를 이용하여 문서의 종류 간의 계층적인 포함 관계와 문서 구조를 표현하였다. 이때 실제로 색인 대상은 트리 구조의 단말 노드(leaf node)에 해당하는 NODE 엘리먼트이다. 색인 정보를 표현할 수 있는 *MODULE* 엘리먼트와 *TEXT*엘리먼트에 대한 정의는

```
:  980000031.sgm
:
<B100>
<B110>1997-008986</B110>
<B120>
<B121>공개특허공보 (B1)</B121>
</B120>
<B190>대한민국특허청 (KR)</B190>
</B100>
<B200>
<B210>1997-010770</B210>
<B220>19970824</B220>
<B240></B240>
</B200>
<B300>
<B310>8711862</B310>
<B320>19950412</B320>
<B330>미국(US)</B330>
</B300>
:
```

(a) SGML 문서 (980000031.sgm)
(a) an SGML document (980000031.sgm)

```
:  980000031.idx
<INSERT>
<RECORD TABLE="파일">
<VALUE FIELD="파일경로">PATDOC\980000031</VALUE>
</RECORD>
<RECORD TABLE="출원">
<VALUE FIELD="출원번호">1997-010770</VALUE>
<VALUE FIELD="출원일자">1997-08-24</VALUE>
<VALUE FIELD="출원종류">공개특허공보(B1)</VALUE>
<SELECTEDVALUE FIELD="파일">
<SELECT FIELD="파일 ID"><FROM>파일</FROM>
<WHERE>파일.파일경로="PATDOC\980000031"</WHERE>
</SELECT></SELECTEDVALUE></RECORD>
<RECORD TABLE="우선권">
<VALUE FIELD="우선권출원번호">8711862</VALUE>
<VALUE FIELD="우선권출원일자">1995-04-12</VALUE>
<VALUE FIELD="우선권주장">1997-008986</VALUE>
<VALUE FIELD="지정국">미국(US)</VALUE>
</RECORD>
:
```

(b) IDDL 명세 (980000031.idx)
(b) an IDDL specification (980000031.idx)

(그림 10) 레코드 생성

(그림 11)과 같다.

```
<!ELEMENT  MODULE  - -      ( TEXT+ )          >
<!ELEMENT  TEXT    - O      EMPTY              >
<!ATTLIST  MODULE  PATH     CDATA   #REQUIRED  >
<!ATTLIST  TEXT    ELEMENT  CDATA   #REQUIRED
           SRC     %URL;    #REQUIRED  >
```

(그림 11) 색인 관련 엘리먼트

(그림 12)는 *MODULE* 엘리먼트를 이용하여 색인 정보를 지정한 예이다. 예를 들어 *INTRO* 엘리먼트에 대한 색인어를 생성하기 위하여 색인 엔진은 파일 경로 "D:\Document\"에 존재하는 엔터티(entity) 파일 intro.ntt을 처리한다.

```
<MODULE PATH="d:\document">
        <TEXT ELEMENT="INTRO" URL="intro.ntt">
        <TEXT ELEMENT="MAIN" URL="main.ntt">
        <TEXT ELEMENT="CLOSE" URL="close.ntt">
</MODULE>
```

(그림 12) 색인 정보 기술

2.2 IDQL

검색 결과의 질을 높이기 위해서는 사용자의 검색 의도를 효과적으로 표현할 수 있는 질의언어가 필요하다. 이에 메타 데이터에 대한 검색과 구조 기반의 내용 검색을 복합적으로 표현할 수 있는 질의 언어인 IDQL을 설계하였다.

SGML 문서의 특정한 엘리먼트의 경우, 그 내용에 자료 유형을 부여하면 적절한 연산의 수행이 가능하다. 특히 관계형 데이터베이스의 자료 유형을 활용하면 검색 속도의 향상 시킬 수 있다. 예를 들어 (그림1)에서 메타 데이터를 나타내는 엘리먼트에 특정한 자료 유형을 부여할 수 있다. 즉, 엘리먼트 *author*에 문자열의 자료 유형을 그리고 엘리먼트 *date*에 날짜 유형을 부여함으로써 자료 유형에 따른 적절한 연산을 수행할 수 있다.

제안된 질의 언어는 메타 데이터의 질의에 사용되는 자료 유형을 숫자(number), 날짜(date), 그리고 문자열(string)로 분류하고, 구조 기반의 색인어 검색을 위하여 텍스트(text) 유형을 추가하였다. (그림 13)은 IDQL의 문법을 BNF(Backaus-Naur Form) 형태로 표현한 것이다.

자료 유형에 따른 연산자의 종류는 SQL의 연산자와 유사하며 날짜 유형의 경우에는 '!=' (같지 않다) 연산자를, 그리고 문자열 유형의 경우 '!=' (같지 않다)와 '$=' (다음 문자열로 시작한다) 연산자를 추가하였다. 한편 텍스트 유형의 경우 '*=' (다음을 포함한다) 연산자를 정의하였다.

따라서 IDQL은 메타 데이터 검색과 구조 기반의 내용 검색을 복합적으로 표현할 수 있다. 예를 들어 1999년 1월 1일 이후에 발표되었으며 제목에 문자열 "SGML"을 포함하는 논문을 찾는 질의문은 다음과 같다.

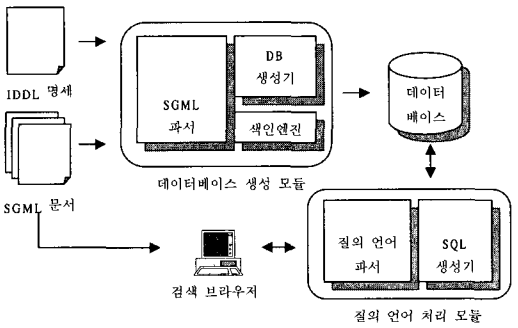
논문지: 발표일자 >= '1999.1.1' and 제목 *= 'SGML'

```
condition ::= dockind ":" logical_factor { logical_op logical_factor }
logical_factor ::= ["("] numeric_field numeric_op number [")"]
        | ["("] date_field date_op quoted_date ) [")"]
        | ["("] string_field string_op quoted_string ) [")"]
        | ["("] text_field text_op quoted_string ) [")"]
number ::= n [ "." [n]]
n ::= digit { digit }
quoted_date ::= " ' " date " ' "
quoted_string ::= " ' " any_character " ' "
logical_op ::= "and" | "그리고" | "or" | "또는"
numeric_op ::= "=" | ">" | "<" | ">=" | "<="
date_op ::= "=" | ">" | "<" | ">=" | "<=" | "!="
string_op ::= "=" | "!=" | "$="
text_op ::= "*="
```

(그림 13) IDQL

3. 검색 도구의 설계 및 구현

본 절에서는 IDDL과 IDQL에 기반하여 개발된 검색 도구의 시스템 구조 및 구성 요소에 대한 보다 자세한 설명을 기술한다. 본 시스템은 구현 환경으로 윈도우 95 환경에서 Visual C++ 5.0와 관계형 데이터베이스 관리시스템을 개발 환경으로 하며 C++ 언어를 사용하였다.



(그림 14) 시스템 구성

3.1 시스템의 구조

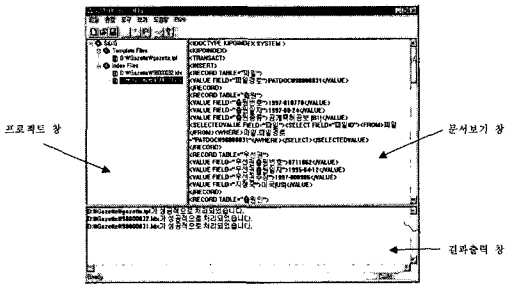
본 논문에서 개발된 검색시스템은 (그림 14)에서 볼 수 있듯이 세 부분으로 구성된다. 즉, IDDL 명세를 처리하여 색인어 데이터베이스를 구성하는 데이터베이스 생성 모듈과 IDQL로 기술된 질의문을 처리하여 검색 결과를 제공하는 질의 언어 처리 모듈로 구성된다. 데이터베이스 생성 모듈과 질의 언어 처리 모듈에 대한 보다 자세한 설명은 다음 절에서 기술한다.

3.2 데이터베이스 생성 모듈

데이터베이스 생성 모듈은 SGML 파서, 데이터베이스 생성기, 그리고 색인 엔진으로 구성된다. 데이터베이스 생성 모듈은 메타 데이터의 저장과 엘리먼트 단위의 색인 및 검색에 대한 정보를 기술한 IDDL 명세와 색인의 대상인 SGML 문서를 입력 받아 데이터베이스를 생성한다. 데이터베이스를 생성하는 과정은 다음의 세 단계로 구성된다.

첫번째는 파일, 엘리먼트, 엘리먼트 이름, 그리고 포함 등의 고정 테이블을 생성한다. 두 번째는 IDDL 명세를 파싱하면서 사용자가 정의한 테이블을 생성하고 필요한 내용을 초기화한다. 세 번째는 실제 문서 내에 포함된 메타 데이터를 데이터베이스에 저장하고 검색 범위로 지정된 엘리먼트를 색인하여 이를 저장한다.

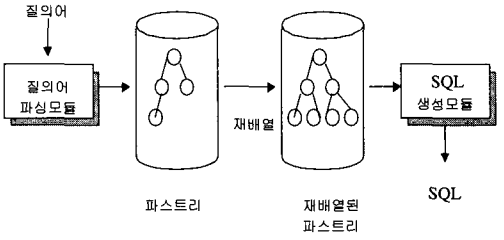
(그림 15)는 데이터베이스 생성 모듈의 실행 화면이다. 본 시스템은 입력데이타의 리스트를 트리 형태로 보여주어 이를 쉽게 추가하거나 삭제할 수 있도록 도와주는 프로젝트 창과, IDDL 명세를 수정 및 저장할 수 있는 문서보기 창, 그리고 처리 결과를 보여주는 결과출력 창으로 구성된다.



(그림 15) 데이터베이스 생성 모듈의 실행 화면

3.3 질의 언어 처리 모듈

IDQL은 메타 데이터에 대한 검색과 구조에 기반한 내용 검색을 불리언(boolean) 연산자를 통하여 복합적으로 표현한다. 이를 관계형 데이터베이스 시스템에 적용하기 위하여 IDQL 질의문을 SQL 문으로 변환하는 과정이 필요하다. 따라서 개발된 질의 언어 처리 모듈은 먼저 IDQL 질의문을 파싱하여 오류의 여부를 검사하고, 파싱의 결과로 생성된 파스 트리(parse tree)를 재배열하여 최종적으로 SQL문을 생성한다. (그림16)은 IDQL을 SQL로 변환하는 과정을 표현한 것이다.



(그림 16) IDQL을 SQL로 변환하는 과정

질의문의 파싱 과정은 스택 연산을 이용한 하향식 파싱 기법을 적용한다. 만일 질의문이 문법에 맞으면 색인어 데이터베이스의 검색필드, 범위, 연산자, 그리고 자료 유형 등을 참조하여 파스 트리를 생성한다. 특히 본 논문은 검색 대상의 이름을 사용자가 재 정의할 수 있도록 하였기 때문에 데이터베이스의 참조 시 이를 고려하여야 한다.

한편 SQL은 상이한 테이블 간의 조인 연산의 결과에 대해서 or 연산을 지원하지 않는다. 예를 들어 메

타 데이터에 대한 검색과 구조 기반의 내용 검색은 파일ID를 계산하기 위하여 조인하는 테이블이 서로 다르다. 따라서 이를 해결하기 위하여 파스 트리를 재배열할 필요가 있다. 한편 논리 연산의 배분법칙을 이용하면 질의문을 최적화하는 효과를 얻을 수 있다. 따라서 or 연산을 사용할 수 없는 문제를 해결하고 질의를 최적화하기 위해서 논리연산의 배분 법칙을 이용하여 or를 상위레벨로 끌어낸다.

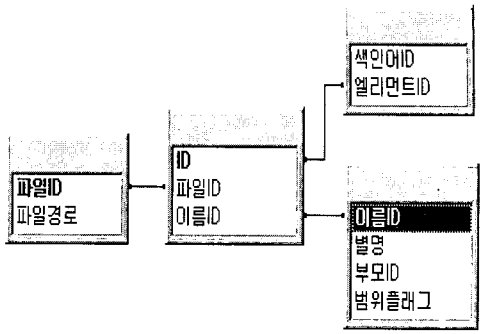
파스 트리의 재배열이 끝난 후, 파스 트리 상의 and나 비교연산자로 시작하는 부 트리(subtree)는 하나의 SQL 문으로 변환된다. 한편 SQL 문에서 여러 테이블을 조인하면 필드 이름 앞에 테이블 이름이 필요하기 때문에 데이터베이스를 참조하여 [테이블 이름].[필드 이름]의 형태로 파스 트리의 내용을 수정한다. 최종적으로 SQL문의 or 관계를 논리합(union) 구문으로 표현한다.

4. 분석 및 고찰

본 논문에서는 실용적인 관점에서 SGML 문서의 메타 데이터와 구조에 기반한 내용 검색을 지원하기 위하여 색인어 데이터베이스를 기술할 수 인어인 IDDL과 질의 언어인 IDQL을 제안하고, 이에 기반한 검색 도구를 개발하였다. 본 절에서는 개발된 검색 도구의 성능을 데이터베이스의 생성과 질의의 측면에서 정성적으로 분석하여 제안된 방법론의 장단점을 기술한다.

먼저 데이터베이스의 생성과 관련하여 제안된 방법론의 특징은 다음과 같다. 일반적으로 SGML 문서를 구성하는 엘리먼트는 계층적인 트리 구조를 형성하며 동일한 색인어가 여러 엘리먼트에 중복될 수 있다. 이러한 SGML 문서의 고유한 특징을 반영하기 위하여 제안된 방법론은 (그림17)과 같이 고정 테이블의 스키마를 정의하였다. *DOCMODEL*에서 기술한 노드는 엘리먼트 이름 테이블의 레코드에 해당한다. 별명은 사상 함수에 의하여 재정의된 이름이고, 부모ID는 계층적인 구조 정보를 표현한다.

문서의 구조나 메타 데이터는 문서의 종류에 따라 다양하다. 이에 메타 데이터를 담는 필드, 테이블, 그리고 테이블 간의 관계 등은 가변적이다. 이와 같은 가변적인 정보는 IDDL에 따라 데이터베이스 생성자에 의하여 기술이 가능하다. 특히 메타 데이터는 검



(그림 17) 고정 테이블의 구조

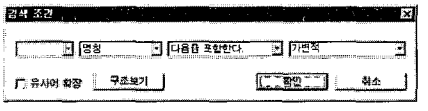
색 대상이 될 수 있는 것이 있고 단순히 검색 결과로 보여지는 것이 있다. 예를 들어, 출판사 이름과 주소에 대한 필드 중에서 출판사 이름만을 검색 대상으로 지정할 수 있다. IDDL은 검색의 대상을 구체적으로 명시할 수 있다. IDDL의 이러한 특징은 특정한 문서처리 환경에 적합한 데이터베이스의 동적인 생성을 가능케 한다. 따라서 이에 기반한 기존의 검색시스템을 재활용할 수 있다는 장점을 갖는다. 또한 SGML로 정의되었기 때문에 데이터베이스 생성자는 별도의 언어를 습득할 필요가 없다.

또한 본 논문에서는 데이터베이스의 생성시, 저장 공간의 효율을 위하여 색인어 테이블을 두 바이트, 네 바이트, 여섯 바이트, 여덟 바이트, 열 바이트, 스무 바이트 등으로 구분한다. 이러한 방법은 결과적으로 색인어의 길이에 따라 해당 테이블을 찾아 조인하기 때문에 검색속도를 향상시킬 수 있다.

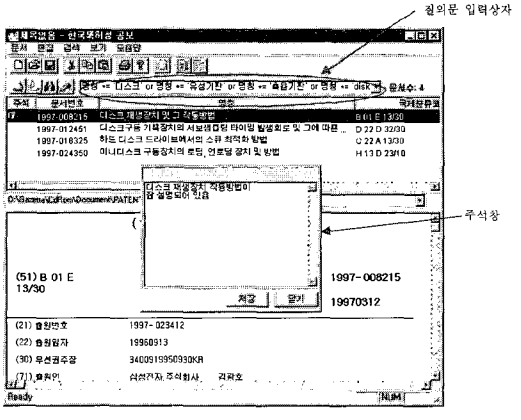
한편, 제안된 IDQL은 메타 데이터에 대한 검색과 구조에 기반한 내용 검색을 복합적으로 지원하도록 설계되었기 때문에 사용자는 하나의 질의문에 두 종류의 검색 조건을 불리언 연산을 통하여 효과적으로 표현할 수 있다. 또한 IDDL 명세에 의하여 생성된 데이터베이스는 사상 함수에 기반하기 때문에 사용자는 해당 문서의 엘리먼트 이름에 독립적으로 검색을 수행할 수 있다.

그러나 제안된 방법론은 SGML 문서의 계층적인 구조의 처리에 있어서 제한적이다. 예를 들어, 문서의 일부분이 "<저자그룹><저자>홍길동</저자><저자>성춘향</저자></저자그룹>"인 문서에 대하여 제안된 방법론은 '저자'가 '홍길동' 또는 '성춘향'인 문서를 검색할 수 있다. 반면에 두 번째 '저자'가 '성춘향'인 문서

에 대한 검색은 지원하지 않는다. SGML 문서의 구조에 기반한 정교한 처리를 지원하기 위해서는 논리적인 구조 정보를 정확히 반영하는 저장 기법이 요구된다. 이에 제안된 방법론은 구조 검색에 있어서 기존 연구와 비교하여 제한적이다. 그러나 실용적인 관점에서 다양한 문서 집합에 대하여 유연한 색인어 데이터베이스의 생성 및 편리한 검색 환경을 제공한다.



(그림 18) 검색 조건의 추가 대화 상자



(그림 19) 사용자 인터페이스

그밖에 본 검색시스템은 데이터베이스를 안전하게 생성할 수 있도록 트랜잭션(transaction) 단위의 처리를 지원한다. 또한 사용자에게 보다 편리한 검색을 지원하기 위하여 유사어 사전을 제공하며 사용자가 질의문을 직접 입력하는 방식과 더불어 (그림 18)과 같이 대화상자를 제공하여 질의를 추가로 입력할 수 있도록 하였다. 이때 시스템은 검색범위나 검색필드에 대한 정보를 데이터베이스를 참조하여 자동으로 제공한다. 그밖에 질의 시 사용자의 편의를 위하여 문서의 구조에 대한 정보를 제공하며 이미 수행된 질의문에 대한 역사(history)를 관리한다. 본 시스템의 사용자 인터페이스는 (그림 19)와 같다.

5. 결 론

본 논문에서는 SGML 문서의 효과적인 검색을 지원하기 위하여 메타 데이터, 검색 범위, 그리고 데이터베이스의 생성에 관련된 정보를 기술할 수 언어로 SGML의 응용인 IDDL을 정의하였다. 또한 메타 데이터 검색과 구조에 기반한 내용 검색을 불리언 연산에 의하여 복합적으로 표현할 수 있는 질의 언어인 IDQL을 설계하였다. 특히 본 연구에서는 IDDL과 IDQL에 기반한 검색 도구를 개발하고 이를 대용량의 문서 집합에 적용하여 제안된 방법론의 성능을 분석하였다.

그 결과, 제안된 방법론의 특징은 다음과 같다. 첫째, IDDL 명세에 따라 데이터베이스를 동적으로 구성하기 때문에 특정한 문서 처리 환경에 적합한 효과적인 데이터베이스의 생성이 가능하다. 둘째, IDDL은 문서의 구조에 대한 사상 함수에 기반하기 때문에 사용자에게 편리한 검색 환경을 제공한다. 셋째, IDDL은 SGML의 응용이므로 데이터베이스 관리자에게 용이한 사용 환경을 제공한다. 넷째, 메타 데이터와 구조 기반의 내용 검색을 복합적으로 표현할 수 있는 질의언어를 지원한다.

부 록

IDDL의 전체 구성은 다음과 같다.

```
<!-- SGML DTD for Description of Index Database                                    -->
<!ENTITY % doctype        "INDEXDB"                                                 >
<!ENTITY % URL            "CDATA"                                                   >
<!ENTITY % datatype       "CHARACTER|INTEGER|LONG|COUNTER|DATE"                     >
<!ENTITY % keywordtype    IGNORED|KEYWORD                                           >
<!ENTITY % bool           "TRUE|FALSE"                                              >
<!ENTITY % uniq           "ONE|DUP"                                                 >
<!ENTITY % req            "REQ|NULL"                                                >
<!-- Entities for special symbols                                                  -->
<!-- &trade &shy and &cbsp are not widely deployed and so not included here        -->
<!ENTITY copy CDATA "&#169"    -- copyright sign                                   -->
<!ENTITY reg  CDATA "&#174"    -- registered sign                                  -->
<!ENTITY amp  CDATA "&#38;"    -- ampersand                                        -->
<!ENTITY gt   CDATA "&#62;"    -- greater than                                     -->
<!ENTITY lt   CDATA "&#60;"    -- less than                                        -->
<!ENTITY quot CDATA "&#34;"    -- double quote                                     -->
<!ENTITY nbsp CDATA " "   -- non breaking space                               -->
<!--          ELEMENTS        MINIMIZATION        CONTENT                         -->
<!ELEMENT     %doctype;       - -                 ( DOCMODEL?,TRANSACT* )         >
<!ELEMENT     DOCMODEL        - -                 ( NODE+ )                       >
<!ELEMENT     NODE            - -                 ( NODE* )                       >
<!ELEMENT     TRANSACT        - -                 ( CREATE* | INSERT* )           >
<!ELEMENT     CREATE          - -                  ( TABLE*, RELATION* )          >
```

```
<!ELEMENT   TABLE          - -   ( FIELD+, INDEX* )                    >
<!ELEMENT   FIELD          - -   ( #PCDATA )                           >
<!ELEMENT   INDEX          - -   ( #PCDATA )                           >
<!ELEMENT   RELATION       - -   ( PRIMARY, FOREIGN)                   >
<!ELEMENT   PRIMARY        - O   EMPTY                                  >
<!ELEMENT   FOREIGN        - O   EMPTY                                  >
<!ELEMENT   INSERT         - -   ( RECORD*, MODULE? )                  >
<!ELEMENT   RECORD         - -   ( VALUE | SELECTEDVALUE )*            >
<!ELEMENT   VALUE          - -   ( #PCDATA )                           >
<!ELEMENT   SELECTEDVALUE  - -   ( SELECT )                            >
<!ELEMENT   SELECT         - -   ( FROM, WHERE )                       >
<!ELEMENT   FROM           - -   ( #PCDATA )                           >
<!ELEMENT   WHERE          - -   ( #PCDATA )                           >
<!ELEMENT   MODULE         - -   ( TEXT+ )                             >
<!ELEMENT   TEXT           - O   EMPTY                                  >
<!--        ELEMENTS       NAME      VALUE       DEFAULT              -->
<!ATTLIST   NODE           NAME      CDATA       #REQUIRED
                           ISRANGE   (%bool;)    TRUE                 >
<!ATTLIST   TABLE          NAME      CDATA       #REQUIRED            >
<!ATTLIST   PRIMARY        TABLE     CDATA       #REQUIRED
                           FIELD     CDATA       #REQUIRED            >
<!ATTLIST   FOREIGN        TABLE     CDATA       #REQUIRED
                           FIELD     CDATA       #REQUIRED            >
<!ATTLIST   FIELD          TYPE      (%datatype;)   CHARACTER
                           SIZE      NUMBER      50
                           NOTNULL   (%bool;)    TRUE
                           KEYWORD   (%keywordtype;) IGNORED          >
<!ATTLIST   INDEX          FIELDNAME CDATA       #REQUIRED
                           ISPRIMARY (%bool;)    TRUE
                           ISUNIQUE  (%uniq;)    DUP
                           ISREQUIRED (%req;)    REQ                  >
<!ATTLIST   RECORD         TABLE     CDATA       #REQUIRED            >
<!ATTLIST   VALUE          FIELD     CDATA       #REQUIRED            >
<!ATTLIST   SELECTEDVALUE  FIELD     CDATA       #REQUIRED            >
<!ATTLIST   SELECT         FIELD     CDATA       #REQUIRED            >
<!ATTLIST   MODULE         PATH      CDATA       #REQUIRED            >
<!ATTLIST   TEXT           ELEMENT   CDATA       #REQUIRED
                           SRC       %URL;       #REQUIRED            >
```

참 고 문 헌

[1] Martin Bryan, 'SGML : An Author's Guide to the Standard Generalized Markup Language', Addison-Wesley, New York, 1988.

[2] Eric van Herwijnen, 'Practical SGML : Second Edition, Kluwer Academic Publishers', Boston, 1994.

[3] Charles F Goldfarb and Yuri Rubinsky, 'The SGML Handbook, Clarendon Press', Oxford, 1990.

[4] Sean McGrath, 'Parseme.1st : SGML for Software Developers', Prentice Hall, 1998.

[5] Martin Bryan, 'SGML and HTML Explained', Addison-Wesley, New York, 1997.

[6] Liora Alschuler, 'ABCD SGML-A User's Guide to Structured information', International Tompson Computer Press, Boston, 1995.

[7] ISO 8879, 'Information processing-Text and office systems-Standard Generalized Markup Language (SGML)', ISO (http://www.iso.ch), 1986.

[8] Heather Brown, Standards for Structured Documents, The Computer Journal, Vol.32, No.6, pp.505-514, 1989.

[9] Ian A.Macleod, A Query Language for Retrieving Information from Hierarchic Text Structures, The Computer Journal, Vol.34, No.3, pp.254-264, 1991.

[10] R. Sacks-Davis, T. Arnold-Moore, and J. Zobel, "Database Systems for Structured Documents," IEICE TRANS. INF. & SYST., Vol.E78 D, No.11, pp.1335-1341, 1995.

[11] 이 택경, 'SGML 문서의 논리적 구조에 근거한 정보검색 시스템에 관한 연구', 연세대학교 석사논문, 1994.

[12] 유 석종, 'SGML 한글문서의 논리적 구조에 근거한 자동색인기법에 관한 연구', 연세대학교 석사논문, 1995.

[13] G. E. Blake, M. P. Consens, P. Kilpelainen, P. A. Larson, T. Snider, and F. W. Tompa, "Text/ Relational Database Management Systems : Harmonizing SQL and SGML," in Proc. of ADB, pp.267-280, Vadstena, Sweden, 1994.

[14] Ian A. Macleod, "Storage and Retrieval of Structured Documents," Information Processing & Management, Vol.26, No.2, pp.197-208, 1990.

[15] V. Christophides, S. Abiteboul, S. Cluet and M. Scholl, "From Structured Documents to Novel Query Facilities," in Proc. of the ACM SIGMOD, pp.313-324, 1994.

[16] V. Christophides, A.Rizk, Querying Structured Documents with Hypertext Links using OODBMS, in Proc. of ECHT, pp.186-197, 1994.

[17] Klemens B hm, Adrian M ller, and Erich Neuhold, "Structured Document Handling a Case for Integrating Databases and Information Retrieval," in Proc. of the ACM CIKM, pp.147-154, Gaithersburg MD, USA, 1994.

[18] Jian Zhang, "Application of OODB and SGML Techniques in Text Database : An Electronic Dictionary System," in Proc. of the ACM SIGMOD, Vol.24, No.1, pp.3-8, 1995.

[19] 김규태, 현득창, 이수연, 정광철 "관계형 데이터베이스를 이용한 SGML 문서 처리", 정보과학회 논문지(C), 제3권, 제3호, pp.238-237, 1995.

[20] 한에노, 박인호, 강현석, 김완석, "SGML 문서의 관리를 위한 객체지향 데이터베이스 설계", 정보처리학회 논문지, 제4권, 제3호, pp.670-684, 1997.

[21] 김현기, 노대식, 강현석, "DTD의존 스키마에 기반한 SGML 문서 저장 시스템 개발에 관한 연구", 정보처리학회 논문지, 제6권, 제5호, pp.1153-1165, 1999.

이 봉 신

e-mail : gromit@datawave.co.kr

1996년 연세대학교 전산과학과 졸업 (학사)

1998년 연세대학교 대학원 컴퓨터과학과(공학석사)

관심분야 : 멀티미디어, 하이퍼미디어, SGML/XML

이 경 호

e-mail : lkh@rainbow.yonsei.ac.kr

1995년 연세대학교 전산과학과 졸업 (학사)

1997년 연세대학교 대학원 컴퓨터과학과(공학석사)

1997년~현재 연세대학교 대학원 컴퓨터과학과 박사과정 재학중

관심분야 : 멀티미디어, 문서처리시스템, 마크업 언어, SGML/XML

고 승 규

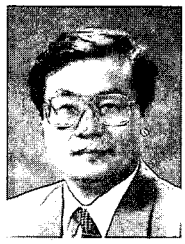
e-mail : pitta@rainbow.yonsei.ac.kr

1992년 연세대학교 전산과학과 졸업 (학사)

1994년 연세대학교 대학원 컴퓨터과학과(공학석사)

1998년 9월~현재 연세대학교 대학원 컴퓨터과학과 박사과정 재학중

관심분야 : 멀티미디어, 정보검색, SGML/XML



최 윤 철

e-mail : ycchoy@rainbow.yonsei.ac.kr

1973년 서울대학교 전자공학과 졸업 (학사)

1975년 6월 Univ. of Pittsburgh (공학석사)

1979년 6월 Univ. of California, Berkeley(공학박사)

1979년 8월~1982년 7월 Lockheed사 및 Rockwell International사 책임연구원

1982년 9월~1984년 1월 Univ. of Massachusetts 연구교수

1984년 3월~현재 연세대학교 컴퓨터과학과 교수

관심분야 : 멀티미디어, 하이퍼미디어, 가상현실, 지리정보시스템