韓國情報敎育學會論文誌
第3卷 第1號 pp.1~12

교육정보화를 위한 문서저장 및 정보검색에 관한 연구

강 무영 · 이 상구

요 약

최근 정보화 도구의 발달로 교육용 소프트웨어와 학습용 DB 등이 빠른 속도로 개발되고 있으며, 인터넷을 통해 보급 및 교환이 활발하게 이루어지고 있다. 또한 교육행정정보화에 따라 여러 교육현장으로부터 생산되는 자료도 매년 엄청나게 불어나게 된다. 본 논문에서는 이러한 대량의 교육정보를 효율적으로 저장하고, 필요한 자료를 검색하여 실시간에 활용할 수 있는 적절한 정보검색시스템과 통합검색 방안에 대해서 제안하였다. 또한 제안한 방법에 의해 간단한 프로토타입을 개발하여 실험하였다.

A Study on Document Storage and Information Retrieval for Educational Informatization

Mu-Yeong Kang · Sang-Gu Lee

ABSTRACT

By recent advances in the information management tools, Education Softwares and Study Wares are rapidly developed and distributed over the Internet very fast. As Education Administrative Information systems are developed, information generated from the education field is growing large and fast. In this paper, we suggest a method of efficiently storing and retrieving such information in the large data basis, in real time. We also demonstrate an information system by the proposed method.

1. 서 론

컴퓨터의 급속한 기술발전으로 여러 분야에서 이를 적용하여 정보화사회로의 환경변화에 대응하고자 많은 노력을 기울이고 있다[1, 5, 10]. 이러한 추세에 따라 교육부에서도 교육정보화촉진시행계획을 마련하고 교육정보화 기반구축을 서두르고 있다. 교육정보화는 초기에 주로 컴퓨터보급, 통신망 구축 등 하드웨어적인 기반시설 구축에 많은 투자를 해왔으며, 아직도 기존 시설에 대한 유지관리, 신규 시설투자에 많은 예산을 필요로 하고 있다.

교육정보화에 있어서 하드웨어적 측면과 더불어 고려하여야할 중요한 요건중의 하나가 소프트웨어

적인 측면이다. 즉 컴퓨터를 잘 활용하기 위해서는 적절한 교육용 소프트웨어의 개발이 필요하고, 여기에 알맞은 교육용 자료를 발굴하여 실수요자인 교육현장에 적시에 보급되어야 한다.

또 교육정보화는 교육정책결정을 지원하기 위한 필수적인 요소로 교육행정정보화를 꼽고 있다. 정보화를 통해 교육현장의 객관적인 자료를 실시간에 수집하고, 이를 분석·가공하여 제공함으로써 정보의 효율성을 극대화하고 정책입안 및 결정에 즉시 반영할 수 있다. '97년 10월에 개발한 교무업무지원시스템 및 문서유통시스템, '98년에 개발한 학교경영업무지원시스템 등은 학교종합정보관리시스템의 서브시스템으로 교육행정정보화를 앞당기기 위해 노력하고 있다[1, 2, 4].

최근 정보화 도구의 발달로 교육용 소프트웨어와 학습용 데이터베이스 등이 빠른 속도로 개발되고 인터넷을 통해 보급 및 교환이 활발하게 이루어지고 있으며, 이러한 교육용 자료가 대량으로 생산되어 정보의 홍수를 이룰 날도 머지 않았다. 이와 함께 교육행정정보화에 따라 많은 교육현장의 자료도 매년 엄청나게 쏟아지게 될 것이다. 따라서 이러한 대량의 정보에 대한 저장·관리와 필요한 자료를 검색하여 실시간에 활용할 수 있는 적절한 시스템의 도입은 교육정보화에 있어서 필연적인 것이다.

본 논문에서는 대량의 정보를 저장하고 관리하는 데 발생하는 문제점과 검색서비스에서의 문제점을 해결할 수 있는 적절한 솔루션을 제시하고자 한다. 특히 저장 및 검색을 위해 연구개발정보센터에서 개발한 KRISTAL-II 시스템을 소개하고, 이에 대한 적용방안도 함께 제시한다. 이를 위해서 2장에서는 효율적인 저장구조와 검색엔진을 탑재한 KRISTAL-II의 시스템구성에 대해서 살펴보고, 3장에서는 KRISTAL-II에 의한 데이터베이스 구축 및 검색서비스 구현방법을 알아본다. 4장에서는 여러 학습자료 생산자와 교육현장에 흩어져있는 정보들을 데이터베이스화하고 이를 통합 검색할 수 있는 방안에 대해서 논의하고, 5장에서 결론 및 향후 연구에 대해서 논한다.

2. KRISTAL-II 검색시스템

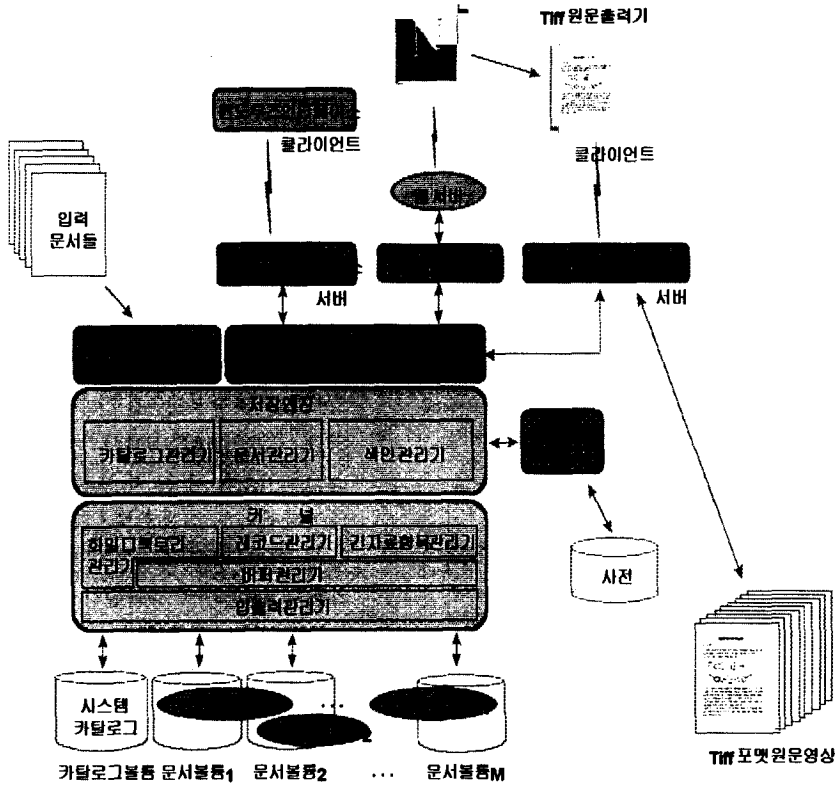
연구개발정보센터(Korea R&D Information Center) 에서는 과학기술정보의 효율적인 유통을 위하여 '93년부터 정보검색시스템인 KRISTAL-II (Korea Research Information of Science and Technology Access Line-II)를 개발하여 왔다.

순수 독자적인 국내 기술로 개발된 KRISTAL-II는 한글, 한자 및 영문이 혼용된 문서를 효율적으로 저장·검색할 수 있는 정보검색시스템으로 과학기술정보유통체제의 기본시스템으로 활용되고 있을 뿐 아니라 국가전자도서관시범사업 등 대형과제의 검색시스템으로 활용되고 있으며, 기타 정부산하 여러 연구소, 대학 등에서 사용함으로써 많은 필드 테스트를 통해 검증된 시스템이다[6, 7, 8, 9].

KRISTAL-II는 NISO Z39.58 프로토콜에 의한 불리언 모델 기반 정보검색시스템으로 초기 버전이 완성되었다. 이것은 서지 정보검색에 적합하도록 가변길이의 필드를 많이 가지고 있는 문서에 대한 검색을 대상으로 하고 있으며, 절단 연산자, 근접도 연산자, 및 히스토리 검색기능을 갖추고 있다. 이후 최대문서크기를 1.3MB로 확장하여 멀티미디어 정보저장을 위한 구조를 갖추었으며, 다중데이터베이스검색, 실시간 정보 추가와 삭제, 검색결과에 대한 랭킹기능 및 색인어 추출을 위한 형태소분석기 등 많은 기능들을 추가하여 보완해 나가고 있다. (그림 1)은 KRISTAL-II의 구성도를 보여주고 있다.

2.1 KRISTAL-II의 커널

커널은 디스크에 자료를 저장하거나 디스크에 저장된 자료를 사용자의 메모리 영역으로 읽어들이는 작업을 제어한다. KRISTAL-II 커널은 입출력 관리기, 버퍼 관리기, 레코드 관리기, 긴 자료항목 관리기를 위해 위스콘신 대학에서 개발한 저장시스템 WISS를 상당 부분 이용하고 있다. 이 저장시스템은 효율적인 자료의 입출력을 위해 유닉스 파일 서브시스템을 우회하여 직접 디스크를 관리한다.



(그림 1) KRISTAL-II 구성도

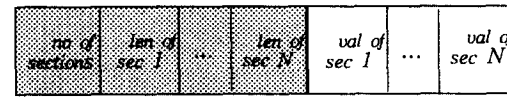
그리고 LRU 방식의 버퍼교환 알고리즘을 사용하여 페이지 단위의 버퍼링을 수행하며, 또한 디스크에 생성된 모든 파일들의 속성정보를 파일디렉토리라는 자료구조를 통해 저장 관리한다.

커널에서 사용자는 레코드와 긴 자료항목의 자료구조를 통해 정보를 저장할 수 있으며, 이들 자료구조는 모두 일련의 바이트 스트림으로 간주되어 처리되며 논리적인 의미는 부여되지 않는다. 레코드는 한 페이지보다 작은 크기의 자료를 유지하며, 긴 자료항목은 한 페이지 이상의 큰 자료를 저장·관리함으로써 레코드의 크기 제한을 보완하고 있다. 즉, 페이지 크기가 4K 바이트일 경우 커널은 최대 1.6M 바이트 크기의 자료저장을 지원한다.

2.2 저장엔진

저장엔진은 하위의 커널을 기반으로 비정형 텍스트 문서들을 저장하고, 이를 판독할 수 있는 기능을 지원한다. 그리고 저장된 텍스트 문서들에 대한 빠른 접근을 지원하기 위해 역파일접근방식의 색인파일을 구현한다.

일반적으로 정보검색의 대상이 되는 문서는 제목, 초록, 본문, 저자 등과 같이 다양한 항목들로 구성된다. 또한 각각의 항목들은 문서마다 서로 다른 크기를 갖는다. 저장엔진의 문서관리기는 이러한 가변적인 크기를 갖는 비정형 텍스트 문서를 저장·관리한다.



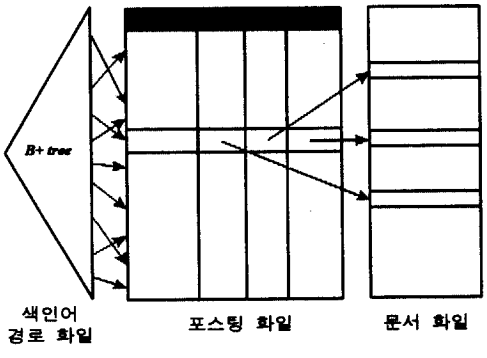
(그림 2) 문서의 저장구조

KRISTAL-II에서 문서는 섹션들의 집합으로 구성된다. (그림 2)는 문서관리기에서 다루는 문서의 논리적 저장 구조를 보여 주고 있다. 문서를 구성

하는 모든 섹션들의 수와 길이 정보가 문서의 헤더에 저장되며, 다음에 섹션들의 실제 내용이 순서적으로 저장된다. 사용자는 레코드 식별자에 의해 저장된 문서들에 접근할 수 있으며, 문서내의 섹션은 섹션번호에 의해 접근할 수 있다.

2.3 색인파일의 구조

(그림 3)은 KRISTAL-II에서의 색인파일의 구조를 보여준다. 그림에서 나타난 바와 같이 색인파일은 색인어 경로파일과 포스팅파일로 구성되어 있다.



(그림 3) 색인파일의 구조

색인어 경로파일은 사용자의 질의에 나타난 단어를 효율적으로 탐색할 수 있도록 전체 문서 파일에 출현한 색인어들을 B+ 트리구조로 관리한다. 즉, 리프노드에 색인어가 저장되며, 루트노드와 내부노드는 이들 색인어를 탐색하기 위한 접근경로를 제공한다. 리프노드는 색인어에 대응하는 포스팅파일 레코드에 대한 포인터를 색인어와 함께 유지한다.

포스팅파일은 색인어들이 출현한 문서들에 대한 정보를 저장하는 파일로서, 색인어마다 하나의 엔트리를 갖는다. 포스팅파일의 엔트리에는 색인어가 출현한 문서식별자와 사용자 질의의 단어간 근접도 연산을 지원하기 위해 문서내의 색인어 위치정보를 저장한다. 이 위치정보는 문서내의 단어번호로 표현되며, 색인어 추출시스템에 의해 자동적으로 부여된다.

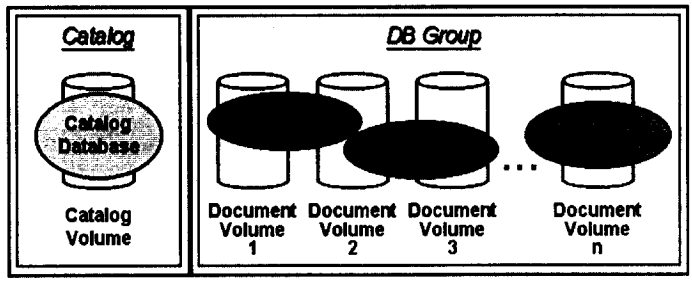
2.4 카탈로그 관리기

카탈로그 관리기는 검색엔진이나 응용프로그램들이 데이터베이스에 접근할 수 있도록 데이터베이스구조, 문서구조, 색인정보 등의 시스템정보를 관리하며, 이들에 접근할 수 있는 기능을 제공한다. 데이터베이스의 시스템정보는 시스템 카탈로그에 저장되며, 시스템 카탈로그는 데이터베이스 그룹 카탈로그, 데이터베이스 카탈로그, 문서파일 카탈로그, 섹션 카탈로그, 결합섹션 카탈로그 등으로 구성된다. KRISTAL-II에서 시스템 관리자는 동일한 스키마구조를 갖는 다중의 데이터베이스를 하나의 데이터베이스 그룹으로 정의할 수 있다. 이러한 기능은 데이터베이스 그룹 카탈로그와 데이터베이스 카탈로그에 의해 지원된다.

KRISTAL-II에서는 데이터베이스 확장을 위해 문서파일 카탈로그를 통하여 여러 개의 볼륨에 분산하여 저장하고 있다. 섹션 카탈로그와 결합섹션 카탈로그는 데이터베이스의 문서를 구성하는 섹션들의 정보를 유지한다. 섹션 카탈로그는 데이터베이스 그룹의 기본적인 섹션들에 대한 섹션이름, 색인방식 등에 대한 정보를 저장하며, 결합섹션 카탈로그는 여러 개의 기본섹션들을 결합해 만들어진 가상 결합섹션들에 대한 정보를 저장한다.

2.5 데이터베이스 관리기

데이터베이스 관리기는 시스템 관리자에 의해 작성된 데이터베이스 스키마정보를 해석하여 카탈로그 관리기를 통해 이 정보를 시스템 카탈로그에 기록한다. 또한 저장 엔진을 이용하여 유닉스파일로서 존재하는 이종의 다양한 원시문서들을 데이터베이스에 적재한다. 적재 과정에서 데이터베이스에 저장되는 문서들은 색인어 추출시스템을 통해 관리자가 결정한 색인 방식에 따라 섹션별로 자동적으로 색인된다. 데이터베이스 관리기는 이들 스키마정보와 적재 정보를 기술하기 위한 별도의 언어를 정의하고 있으며, 이 언어를 실행하기 위한 인터프리터를 사용한다.



(그림 4) KRISTAL-II 데이터베이스 구조

2.6 검색 엔진

부울모델에서 문서는 색인어들의 집합으로 표현되고, 질의는 색인어들을 부울연산자 AND, OR, NOT으로 연결한 부울수식이며, 검색되는 문서는 질의로서 주어진 부울수식을 만족하는 문서들이다. KRISTAL-II의 검색 엔진은 부울모델을 지원하며, 카탈로그 관리기와 저장엔진을 이용하여 사용자질의를 만족하는 문서들을 데이터베이스로부터 검색한다.

국제 표준화 기구 중의 하나인 National Information Standards Organization (NISO)에서는 1991년에 Z39.58-199x를 발표하였다. 이것은 온라인 정보검색을 위한 사용자명령을 표준화한 것으로서, Z39.58-199x의 FIND 명령어는 부울모델을 근간으로 하고 있다. KRISTAL-II에서 사용자 질의의 구문형식은 Z39.58-199x의 FIND 명령어를 기초로 만들어졌으며, 부울연산(AND, OR, NOT), 근접도연산(NEAR, WITHIN), 절단연산(*, ?), 관계연산(<, <=, >, >=, =, -), 사용자질의의 히스토리 연산(SET), 검색결과의 정렬, 다중검색 범위지정을 통한 검색 등과 같은 다양한 연산을 제공한다.

2.7 색인어 추출 시스템

정보검색에서 자동색인은 문서의 내용을 대표할 수 있는 색인어를 추출하는 것을 말하며, 일반적으로 색인어 추출방법은 정보검색시스템의 검색효과에 중요한 영향을 미치는 것으로 알려져 있다. 현재 KRISTAL-II는 데이터베이스에 대한 섹션별 색인을 지원하며, 색인어 추출방식에 따라 섹션단위, 어절단위, 형태소단위의 색인방법을 제공한다. 섹션단위의 색인은 섹션 값 전체를 하나의 색인어로 선정하는 방식으로, 섹션 값에 대한 완전일치 검색을 지원한다. 어절단위의 색인은 각 섹션에서 색인어로서 가치가 없는 불용어를 제외한 모든 어절들을 원문에 나타난 형태 그대로 색인어로써 추출한다. 그리고 형태소단위의 색인은 한글문장에 대해 형태소분석을 수행하여 모든 어절들을 명사, 조사, 부사 등의 형태소단위로 분리한 후, 불용어들을 제거하고 색인어로서 의미가 있는 단순명사들을 색인어로 추출한다. KRISTAL-II의 색인어 추출시스템은 한글문장의 분석을 위해 연구개발정보센터에서 개발한 형태소분석기를 이용하고 있다.

3. 데이터베이스 구축 및 검색서비스 구현

KRISTAL-II는 (그림 4)와 같이 목록 데이터베이스(Catalog Database)와 문서 데이터베이스(Document Database)들로 구성된다. 목록 데이터베이스는 목록볼륨(Catalog Volume)에 저장되고, 문서 데이터베이스들은 대용량 데이터베이스를 처리하기 위해 하나 또는 여러 개의 문서 볼륨(Document Volume)에 분산되어 저장된다. 문서 데이터베이스는 구축하려는 실제의 문서들과 그 문서들에 대한 색인을 저장하며, 목록 데이터베이스는 데이터베이스들에 대한 문서구조, 저장위치, 색인방법 등에 대한 정보들을 저장하여 관리한다.

KRISTAL-II에서는 다중볼륨(Multi-volume)에

의해 자료분산 기능을 지원한다. 분산된 문서 데이터베이스 중에서 동일한 저장장치에 저장되는 문서들의 집합은 문서파일(또는 문서그룹)이라고 불리우며, 각 문서파일에 대한 색인파일은 문서파일과 동일한 저장장치에 생성된다.

KRISTAL-II는 여러 개의 단위 문서 데이터베이스들을 하나로 묶어서 데이터베이스 그룹(Database Group)으로 확장할 수 있고, 데이터베이스의 레코드를 식별하기 위해 각 레코드마다 물리적인 문서식별자(Record Identifier, RID)를 부여한다. 물리적 문서식별자는 데이터베이스 그룹 전체에서 유일하도록 시스템에 의해 자동으로 부여되며, 볼륨식별자, 페이지식별자, 페이지 안에서의 슬롯번호등 총 8자의 숫자로 구성된다. 또한 KRISTAL-II는 문서의 빠른 검색을 지원하기 위해 역파일(Inverted File) 구조의 색인어 저장방식을 제공한다. 이 방식은 대용량의 저장공간을 필요로 하는 단점을 지니고 있다. 따라서 이러한 색인파일의 오버헤드를 감소시키기 위하여 8자리의 숫자로 구성된 물리적 문서식별자를 사용하는 대신 논리적 문서식별자를 사용하였으며, 이 논리적 문서식별자로 DOCLOC.SYS라는 파일을 이용하였다. DOCLOC.SYS 파일은 논리적 문서식별자를 물리적 문서식별자로 치환시키는 역할을 담당하며, 목록 데이터베이스와 동일한 디렉토리 내에 저장된다.

KRISTAL-II의 문서구조는 기본섹션(Primitive Section)과 결합섹션(Complex Section)으로 분류된다. 기본섹션(Primitive Section)은 하나의 문서를 구성하는 최소단위의 문서 구성요소이며, 검색과정에서 문서의 검색범위를 지정하기 위한 기본단위로 사용한다. 또한 문서색인을 위한 단위로 사용되기도 한다. 결합섹션(Complex Section)은 여러 개의 기본섹션들을 결합하여 논리적으로 하나의 의미를 갖는 새로운 섹션을 구성한 것이다. 이것은 실제값을 저장하지는 않고 사용자의 관점에서 하나의 섹션단위로 인식되는 가상섹션이다.

또한 문서색인은 검색속도와 검색효과에 중요한 영향을 미치는 요인으로 KRISTAL-II에서는 <표 1>과 같이 다양한 색인방법을 지원하고 있다. 따라서 데이터베이스 설계자는 문서의 기본섹션마다 적절한 색인방법을 지정함으로써 검색시스템의 성능과 질을 크게 향상시킬 수 있다.

<표 1> 색인방식별 사용연산자

(O : 연산자 사용 가능, X : 연산자 사용 불가능)

색인방식	부울 연산자	근접도 연산자	절단 연산자	관계 연산자
EXACT	O	O	O	X
INC_NONE	O	O	O	X
INC_MA	O	O	O	X
INC_CHAR	O	O	O	X
FLOAT	X	X	X	O
INTEGER	X	X	X	O
STRING	X	X	O	O
EXACT_MA	O	O	O	X
HANJA_EXACT_MA	O	O	O	X
HANJA_INC_NONE	O	O	O	X
HANJA_INC_MA	O	O	O	X

1. 부울 연산자 : AND(&), OR(|), NOT(!)
2. 근접도연산자 : /W, /N
3. 절단 연산자 : *, ?
4. 관계 연산자 : <, <=, >, >=, =, -

3.1 데이터베이스 구축

본 논문에서는 위의 데이터베이스 구조에 의한 데이터베이스 구축 및 검색서비스에 대한 구현 사례로 초등학교 학교생활 평가관리의 간략한 예를 통해 알아보기로 한다.

3.1.1 데이터베이스 설계

현재 초등학교의 새 교육과정은 어린이들의 특성에 알맞도록 구성하여 전인적인 인간을 육성하는데 역점을 두고 있다. 따라서 평가방법도 지필평가를 지양하고 학생 스스로 산출물을 만들거나 행동으로 나타내며 답을 작성하도록 요구하는 수행평가 방법을 택하여 평가하고 있다. 이러한 취지 하에 학교생활에 대한 가정 통지표를 종전의 정량적 평가표에서 정성적 평가로 바뀌고 있다. 이러한 자료를 데이터베이스로 구축하여 검색하기 위해서는 정

성적 평가내용에 대한 적절한 색인어 추출과 질의를 통하여 원하는 정보를 획득할 수 있다. 이것을 위한 초등학교 통지표에 대한 데이터베이스를 구축할 때 필요한 데이터베이스 설계를 기본섹션과 결합섹션에 의해 표현할 수 있다.

(1) 기본섹션의 정의

기본섹션은 <표 2>와 같이 각 섹션을 대표하는 섹션명, 섹션에 대한 설명, 각 섹션에 대한 색인방법을 표현한다.

<표 2> 기본섹션 정의

섹션명	설명	색인방법
AN	고유번호	STRING
NM	학생이름	EXACT
TM	교장선생님	EXACT
SM	교감선생님	EXACT
RT	담임선생님	EXACT
SL	바른생활	INC_NONE
KO	국어	INC_NONE
MA	수학	INC_NONE
IL	슬기로운생활	INC_NONE
HL	즐거운생활	INC_NONE
LO	교과학습종합의견	INC_MA
TO	협의활동	INC_NONE
RO	역할분담활동	INC_NONE
FR	친교활동	INC_NONE
EV	행사활동	INC_NONE
SO	특별활동종합의견	INC_MA
RU	준법성	INC_NONE
DI	근면성	INC_NONE
RE	책임감	INC_NONE
CO	협동성	INC_NONE
SE	자주성	INC_NONE
LB	생활습관	INC_NONE
BO	행동발달종합의견	INC_MA
HI	키	FLOAT
WE	몸무게	FLOAT
CH	가슴둘레	FLOAT
SH	앉은키	FLOAT
AT	출결사항	NOT_EXIST

(2) 결합섹션의 정의

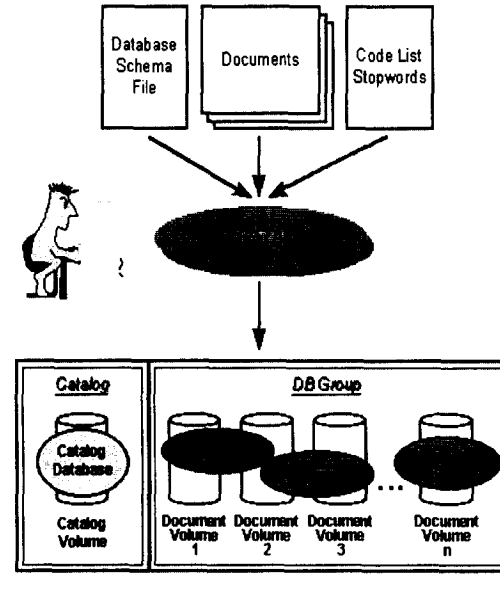
결합섹션은 <표 3>과 같이 각 섹션을 대표하는 섹션명, 섹션에 대한 설명, 각 섹션을 구성하고 있는 기본섹션의 리스트로 표현한다.

<표 3> 결합섹션 정의

섹션명	설명	기본섹션리스트
BI	기본검색필드	AN, NM, RT
LPS	교과학습발달상황	SL, KO, MA, IL, HL, LO
SAS	특별활동상황	TO, RO, FR, EV, SO
BPS	행동발달상황	RU, DI, RE, CO, SE, LB, BO

3.1.2 데이터베이스 적재

(그림 5)는 KRISTAL-II 데이터베이스를 적재하는 과정을 개념적으로 보여주고 있다. 이와 같은 과정 중 상위부분이 데이터베이스 적재를 위해 관리자가 준비해야 할 파일이다.



(그림 5) 데이터베이스 적재 과정

데이터베이스 생성 및 문서 적재를 위한 명령어들은 (그림 6)과 같이 8개가 있다.

1. **DATABASE_DIRECTORY**
   데이터베이스의 구축을 위해 필요한 문서 데이터베이스와 목록 데이터베이스(CATALOG.SYS), 매핑파일(DOCLOC.SYS)등을 저장하기 위한 디렉토리 정의

2. **CREATE_DOCUMENT_VOLUME**
   문서볼륨의 생성 및 초기화를 위한 관련 정보들을 정의

3. **CREATE_SCHEMA**
   기본섹션 및 결합섹션의 정의, 기본섹션의 색인방식을 지정

4. **CREATE_DATABASE**
   데이터베이스 그룹에 속하는 데이터베이스들을 정의

5. **DEFINE_DOCUMENT_STRUCTURE**
   각 섹션을 적재할 수 있도록 하는 문서구조에 대한 정의

6. **DEFINE_DOCUMENT_GROUP**
   각 문서 데이터베이스로 적재되는 원시문서들의 그룹에 대한 정보를 정의

7. **LOAD_DATABASE**
   정의된 문서 구조에 맞추어 원시문서들을 데이터베이스내에 적재하고, 각각의 기본섹션을 지정된 색인방법에 따라 색인할 것을 KRISTAL에게 지시

8. **END**
   데이터베이스 적재 및 색인작업이 완료

(그림 6) 스키마정의를 위한 KRISTAL 명령어

(1) 데이터베이스 스키마파일 (Database Schema File)

데이터베이스 디렉토리와 문서볼륨, 데이터베이스 그룹, 섹션정의, 기본섹션의 색인방식, 원시문서의 구조, 원시문서로부터 데이터베이스로의 적재방법 등에 대한 정보를 기술한다. (그림 7)은 성적관리용 스키마파일의 예를 보여주고 있다.

(2)원시문서파일 (Documents)

데이터베이스에 적재될 원시문서들로, 이질구조를 가진 문서도 데이터베이스 스키마파일에 기술되어 있는 원시문서의 구조를 통하여 하나의 데이터베이스 내에 적재가 가능하다. (그림 8)은 성적관리용 원시문서 파일의 예를 보여주고 있다.

```
DATABASE_DIRECTORY=/home/SCHOOL/volume
CREATE_DOCUMENT_VOLUME{
    (1) VOLUME_NAME='SCHOOLVOL'
        NUMBER_OF_EXTENTS=28000
        EXTENT_SIZE=16
};
CREATE_SCHEMA{
    DATABASE_GROUP_NAME=SCHOOLGR
    SECTION_DEFINITION{
        (1)  LABEL="고유번호"
             SECTION_NAME=AN,
        (2)  LABEL="학생이름"
             SECTION_NAME=NM,
             ...(중략)
        (12) LABEL="협의활동"
             SECTION_NAME=TO,
             ...(중략)
    }
    COMPLEX_SECTION_DEFINITION{
        (1)  LABEL="기본검색필드"
             SECTION_NAME=BI
             SECTIONS=(AN, NM, RT)
    }
    INDEX_DEFINITION{
        (1)  SECTION_NAME=AN
             INDEX_TYPE=STRING,
        (2)  SECTION_NAME=NM
             INDEX_TYPE=EXACT,
             ...(중략)
        (11) SECTION_NAME=LO
             INDEX_TYPE=INC_MA
             S T O P W O R D =
('/home/SCHOOL/dict/stopwords'),
             ...(중략)
    }
};
CREATE_DATABASE{
    (1) DATABASE_GROUP_NAME=SCHOOLGR
        DATABASES=(SCHOOLDB)
};
DEFINE_DOCUMENT_STRUCTURE{
    DATABASE_GROUP_NAME=SCHOOLGR

STRUCTURE_DEFINITION=DOCUMENT_STRUCTURE_TYPE{
        (1)  TAG="@Record_View"
             ACTION=DISCARD
             NEW_DOCUMENT_FLAG=TRUE,
        (2)  TAG="#AN="
             ACTION=COPY
             SECTION_NAME=AN,
        (3)  TAG="#NM="
             ACTION=COPY
             SECTION_NAME=NM,
             ...(중략)
        (13) TAG="#TO="
             ACTION=TRANSLATE
             METHOD=SUBSTRING(1,1) WITH
                    '/home/SCHOOL/code/SA_CODE'
             SECTION_NAME=TO,
             ...(중략)
    }
};
DEFINE_DOCUMENT_GROUP{
        (          1          )
SCHOOLSRC=('/home/SCHOOL/data/SCHOOL.dat')
};
LOAD_DATABASE{
    (1) FROM=SCHOOLSRC
        TO=(DATABASE_GROUP_NAME:SCHOOLGR
            DATABASE_NAME:SCHOOLDB
            VOLUME_NAME:'SCHOOLVOL'
            FILE_NAME:SCHOOLFN)
            WITH=DOCUMENT_STRUCTURE_TYPE
};
END
```

(그림 7) 성적관리용 스키마파일

```
@Record_View
#AN=199901010307
#NM=홍길동
#TM=김교장
#SM=김교감
#RT=김담임
#SL=학교와 가정의 생활이 다름을 알고 학교규칙을 지키려고 노력함
#KO=겪은 일 중에서 중요한 일을 선택하여 그림일기로 나타낼줄 알고 있음
#MA=두자리수에서 10개씩 묶음수와 낱개의 구성관계를 잘 알고 있음
#IL=자연에 대한 관심과 이해를 잘하며 특히 여름철 기후를 알맞게 표현함
#HL=달리기를 잘하여 모든 운동에서 뒤어나게 잘하고 있음
#LO=각 교과성적이 고루 우수하며 특히 체육과에 소질이 있고 취미가 많음
#TO=A
#RO=B
#FR=C
#EV=A
#SO=당번활동을 능동적으로 열심히 잘함
#RU=A
#DI=A
#RE=B
#CO=C
#SE=A
#LB=A
#BO=책임감이 강하고 부지런하며 매사에 활동적임
#HI=134.6
#WE=29
#CH=62
#SH=71.5
#AT=출석일수 220일중 결석 1회, 지각 3회
@Record_View
...
```

(그림 8) 원시문서 파일
(SCHOOLSRC=/home/SCHOOL/data/SCHOOL.dat)

(3) 코드 목록(Code List)

데이터베이스에 적재할 때 실제값으로 변경해야 할 내용을 저장한 것으로, (그림 9)는 성적표에서 관리코드를 예시한 것이다.

```
A='우수함'
B='보통'
C='노력요함'
```

(그림 9) 코드파일
(/home/SCHOOL/code/SA_CODE)

(4) 불용어 목록(Stopwords)

색인어를 추출할 때 색인에서 제외시킬 용어를 저장한 파일이다.

(5) 사용자 정의 사전(Indexwords)

형태소를 분석할 때 사용하는 사용자 정의 사전을 저장한 파일이다.

이와 같이 파일들이 준비되면 KRISTAL을 통해 데이터베이스에 적재하게 된다. KRISTAL은 데이터베이스를 적재하기 위해 사용하는 프로그램으로 데이터베이스 적재에 필요한 모든 정보를 데이터베이스 스키마파일로부터 가져온다.

KRISTAL은 컴퓨터의 표준입력(standard input)을 통해 입력된 관리자의 명령어를 수행하는 일종의 명령어 해석기(command line interpreter)로서, 관리자는 이 프로그램에게 데이터베이스 생성 및 문서적재를 위한 명령어를 입력할 수 있다. (그림 10)은 KRISTAL 명령어 사용법을 보여주고 있다.

```
1) KRISTAL -Syntax < Schema_File_Name
   스키마파일의 문법(syntax)의 오류검사
2) KRISTAL -Create < Schema_File_Name
   스키마파일에 정의된 목록볼륨과 문서볼륨을 생성
3) KRISTAL -Semantic < Schema_File_Name
   문서구조에 기술된 섹션이름, 코드목록 파일명, 원시문서 그룹에 대한 원시문서명등에 대한 존재 여부 확인과 데이터베이스 그룹 이름, 문서볼륨 이름, 문서구조 이름등을 확인
4) KRISTAL -ALoad < Schema_File_Name
   원시문서를 읽어들여 문서 데이터베이스에 적재하고, 적재된 문서로부터 색인어를 추출하여 색인파일에 적재
5) KRISTAL -Append < Schema_File_Name
   문서볼륨이나 데이터베이스를 추가하여 생성
```

(그림 10) KRISTAL 명령어 사용법

3.2 검색서비스 구현

KRISTAL-II의 검색엔진을 이용하여 사용자의 질의를 처리하기 위해 개발자는 검색엔진 KRISTAL-FIRE의 응용프로그램 개발용 사용자 인터페이스(API)를 이용해야 한다.

KRISTAL-FIRE는 NISO의 Z39.58을 따르는 부울질의어 처리를 지원하며, 여러 개의 응용 프로그램 인터페이스를 C 언어로 작성된 라이브러리(library)의 형태로 제공한다. (그림 11)은 이들 API에 대한 리스트이다.

4. 통합검색을 위한 서비스방안

교육정보에 대한 통합검색을 위해서는 정보검색

FIRE_Initialize (CatDir)
검색엔진을 기동

FIRE_Terminate (CatDir)
검색엔진을 종료

FIRE_OpenDBs (DBGrpName, DBCnt, DBNames)
데이터베이스 그룹과 그룹내에 포함되어 있는 일련의 데이터베이스들에 대한 개방

FIRE_CloseDBs ( )
현재 개방되어 있는 데이터베이스 그룹과 그룹내의 데이터베이스들에 대한 사용을 종료

FIRE_FindDoc (Where, SetNum, MemCnt)
사용자의 부울질의를 만족하는 문서들을 검색하고, 검색된 문서들의 식별자리스트를 검색결과 집합에 저장한 후, 결과집합의 집합번호(SetNum)와 검색된 문서들의 수(MemCnt)를 반환

FIRE_Sort (SecName, SetNum, MemFrom, MemTo, Flag, SortBuf, NumIDs)
검색된 문서 결과집합에서 사용자가 지정하는 범위의 부분 결과집합을 정렬

FIRE_GetSecLen (SetNum, MemNum, SecName, Len)
검색 결과집합에서 한 문서에 대하여 SecName에서 지정한 섹션의 데이터 길이를 읽어옴

FIRE_ReadSecVal(SetNum, MemNum, SecName, Start, Len, SecBuf)
검색된 문서들의 실제 내용을 읽어오기 위해 사용

FIRE_GetDocIDList(SetNum, DocIDList)
문서검색 결과집합에 저장되어 있는 문서식별자들을 접근하기 위해 제공되는 모듈

FIRE_GetSecLenDID (DocID, SecName, Len)
DocID에서 지정한 문서에 대하여 SecName에서 지정한 섹션의 데이터 길이를 읽어옴

FIRE_ReadSecValDID (DocRID, SecName, Start, Len, SecBuf)
문서식별자를 직접 사용하여 섹션의 내용을 읽어옴

FIRE_ClearSet(SetNum)
SetNum에 의해 지시되는 검색 결과집합을 시스템으로부터 제거

FIRE_ClearSetsAll( )
시스템내에서 유지되고 있는 모든 검색 결과집합들을 제거

FIRE_FindTermPrev(SecName, SearchKey, Cnt, TermListBuf, RetCnt)
색인어를 사용자가 조회할 수 있도록 하기 위해 제공되는 모듈

FIRE_FindTermNext(SecName, SearchKey, Cnt, TermListBuf, RetCnt)
색인어를 사용자가 조회할 수 있도록 하기 위해 제공되는 모듈

FIRE_FirstDoc(DocID)
검색중인 데이터베이스의 첫번째 문서식별자를 읽어옴

FIRE_LastDoc(DocID)
검색중인 데이터베이스의 마지막 문서식별자를 읽어옴

FIRE_NextDoc(CurDocID, NextDocID)
현재 문서식별자(CurDocID)로부터 다음 문서식별자(NextDocID)를 읽어옴

(그림 11) KRISTAL-FIRE API(1)

FIRE_PrevDoc(CurDocID, PrevDocID)
현재 문서식별자(CurDocID)로부터 이전 문서식별자(PrevDocID)를 읽어옴

FIRE_GetOccCnt(SecName, IdxTerm, OccCnt)
색인어가 검색중인 데이터베이스 그룹의 모든 문서 한 섹션에서 출현한 횟수를 읽어옴

FIRE_GetSecNames(SecName, SecNameList, SecCnt)
섹션 이름에 따라서 복합섹션이면 복합섹션을 구성하는 섹션들을 모두 읽어옴

(그림 11) KRISTAL-FIRE API(2)

인터페이스 프로토콜 ANSI/NISO Z39.50 - 1995 Information Retrieval: Application Service Definition and Protocol Specification에 입각한 연계모듈을 개발하여 타 Z39.50서버와 접속함으로써 로컬시스템과 외부시스템을 하나의 시스템처럼 통합하여 검색할 수 있다.[9] 정보수요자가 로컬시스템이 아닌 타 시스템으로부터 인터넷에 공개된 자료에 대해 원하는 정보를 손쉽게 획득하는 것은 쉬운 일이 아니다. 대부분 시스템마다 서로 다른 사용자 인터페이스 환경을 제공하고 있어서 그들 시스템의 독자적인 이용방법을 충분히 숙지해야만 원활한 정보검색을 수행할 수 있기 때문이다.

이러한 다양한 환경에서도 원활한 정보검색을 할 수 있도록 문제를 해주는 것이 Z39.50 프로토콜에 의한 인터페이스를 제공하는 것이다. Z39.50은 초기에 서지 데이터(bibliographic data)를 관리하는 시스템으로 활용되었지만 아주 범용적이고 확장성이 용이해 현재 급격히 변화하고 있는 네트워크 정보환경에 맞도록 꾸준히 성장·발전하고 있다. (그림 12)는 Z39.50-1995에서 제공되는 주요기능을 나열한 것이다.

정보검색엔진과 Z39.50 서버시스템은 크게 두 가지로 연동할 수 있다. 하나는 Z39.50 프로토콜의 기능과 검색엔진을 분리하여 서버내 검색엔진 인터페이스를 통해 서로 다른 검색엔진들과 연동하도록 설계하는 방법이고, 다른 하나는 Z39.50 프로토콜의 기능과 검색엔진을 결합하여 Z39.50 서버에서 직접 정보를 검색하도록 설계하는 방법이다.

전자는 Z39.50 서버가 다양한 검색엔진과 연동하기가 편하다는 장점을 갖고 있지만 서버 내에 검색엔진 인터페이스를 별도로 만들어야 하는 비용을

1. **Initialization(Init)**
   클라이언트가 기존의 TCP접속의 최상위에 Z39.50세션 설정을 요청하면서 버퍼의 크기, 사용가능한 기능들, 버전등을 제시하고 이용자의 신분정보을 제공
2. **Access Control**
   서버가 클라이언트의 요구를 일시 정지하고 클라이언트에게 신분확인을 위한 정보 요구
3. **Search**
   클라이언트가 탐색을 요청하여 서버에 검색 결과집합을 생성
4. **Retrieve**
   - Present : 클라이언트가 검색 결과집합으로부터 특정 레코드를 요청
   - Segmentation : 전체 레코드 크기가 미리 정해진 전송크기를 초과할 경우 나누어 전송
5. **Result-Set-Delete**
   클라이언트가 한 개 이상의 검색 결과집합을 삭제 요청
6. **Accounting/Resource Control Facility**
   -Resource Control : 정보이용량에 대한 확인을 요구
   - Trigger-Resource-Control : 클라이언트가 서버에게 동작중지 요청이나 서버가 클라이언트에게 Resource Control을 요구하게 함
   - Resource-Report : 클라이언트가 요구하며 자원 사용량에 대한 정보 요청
7. **Browse/Scan**
   이용가능한 데이터베이스, 지원되는 속성, 레코드 구문구조, 엘리먼트 사양 등 특정 서버에 대한 정보를 찾음
8. **Extended Service**
   추후 사용을 위한 검색 결과집합의 저장, 주기적인 탐색질의 수행, 출력지시 등 서버에게 요청할 수 있는 작업
9. **Proximity Searching**
   클라이언트가 인접탐색을 지정할 수 있는 질의 유형
10. **Sort**
    검색 결과집합을 정렬
11. **Termination**
    세션 종료 요청
12. **새로운 레코드 구문구조**
    - 단순 비구조화 텍스트 레코드 구문구조
    - OPAC 레코드 구문구조
    - 일반 레코드 구문구조

(그림 12) Z39.50-1995에서 제공되는 주요기능

필요로 한다. 후자는 검색엔진 인터페이스를 별도로 만들 필요가 없지만 다른 검색엔진과의 연동에 불편함이 따른다. 국가 주요전자도서관 연계사업의 일환으로 구축된 시스템[9]은 전자의 경우를 따라 개발되어 타 서버와 통합검색이 원활히 이루어지고 있다.

5. 결론 및 향후 연구

본 논문에서는 정보화사회로의 환경변화가 가속화되고 있는 현 시점에서 교육정보화 기반구축의 일환으로 교육정보를 효율적으로 저장하여 필요한 정보를 시간과 공간의 제약없이 실시간에 획득할 수 있는 정보검색시스템에 대한 솔루션을 제안하였다.

이를 위해 연구개발정보센터에서 개발한 정보검색시스템 KRISTAL-II를 소개하였고, 초등학교 학교생활 평가관리를 위한 간략한 데이터베이스 구축에 대해서 살펴보았다. 또한 여러 교육 현장에서 발생하는 교육정보를 Z39.50 프로토콜에 의해 통합검색을 할 수 있는 방안에 대해서 논의하였다.

요약하면 정보검색시스템 KRISTAL-II는 대용량의 데이터베이스로부터 사용자의 질의를 만족하는 문서들의 검색을 지원한다. 이 시스템은 NISO에서 제정 발표한 질의 구문 형식을 구현함으로써 다양한 질의 기능들을 제공한다. 저장엔진은 정보검색의 주된 대상이 되는 가변길이의 비정형 텍스트 문서들에 대한 저장과 접근방법을 제공하며, 카탈로그 관리기는 서비스 제공자로 하여금 대용량의 데이터베이스를 디스크 용량의 한계를 넘어 분산 저장할 수 있다. 색인어 추출시스템은 다양한 색인방식뿐만 아니라 검색효과의 개선을 위한 색인어 후처리도 지원한다. 이러한 방법으로 지역별 데이터베이스를 구축하여, Z39.50 프로토콜에 의해 여러 지역에 분산되어 있는 교육정보를 한꺼번에 통합검색할 수 있다.

향후 연구과제로서 정보검색시스템 KRISTAL-II의 성능의 개선에 노력을 기울이고, 이를 교육정보화 기반구축에 응용할 수 있는 방안에 대해 연구하고자 한다. 현재 대용량 문서에 대한 효과적인 색인방법과 온라인 색인방법에 대해 연구하고 있다. 색인기법은 문서의 효과적인 저장과 더불어 정보검색에 있어서 대단히 중요한 기능이다. 이러한 색인기법은 문서의 양에 따라 다르지만 필연적으로 대량의 시스템자원을 필요로 하며 많은 시간이 소요된다. 따라서 문서 내에 있는 모든 단어들에 대해 색인하기 위해서는 저장공간과 소요시간을 최소화하는 방법에 대한 연구가 절실히 요구되어 지금까지 여러 연구자들에 의해 많은 연구가 이루어져 왔다[3, 11, 12, 13, 14, 15]. 이러한 색인방법들에 대한 비교분석과 개선된 색인방법을 연구하여 적용함으로써 KRISTAL-II의 성능을 개선하고자 한다.

참 고 문 헌

[1] 교육부(1997), 교육부정보화촉진시행계획, 교육부 자료
[2] 이무근(1999), 21세기 지식기반사회 구현을 위한 인적자원 개발과제, 교육부 해외교육정보 확산을 위한 세미나 주제발표 자료
[3] 이상구(1997), 대용량 문서를 위한 인덱스 생성기법에 관한 연구, 연구개발정보센터 연구과제 최종보고서
[4] 이옥화(1998), 교육정보화의 현황과 과제해결을 위한 종합적 접근, 컴퓨터교육학회논문지, Vol. 1, No. 1, Jun. 1998
[5] 이현구(1997), 국가전자도서관구축기본계획, 국회도서관, 국립중앙도서관, 법원도서관, 산업기술연구원, 연구개발정보센터, 첨단학술정보센터 공동 발간자료
[6] 연구개발정보센터(1995), 정보검색을 위한 효율적인 저장시스템 개발(Ⅰ), 과학기술부 특정연구과제 보고서
[7] 연구개발정보센터(1997), 전자도서관을 위한 KRISTAL-II 성능 개선, 현대정보기술(주)의 KRISTAL-II 사용권 및 Customizing에 대한 최종보고서
[8] 연구개발정보센터(1998), 과학기술정보유통체제 구축사업(Ⅷ), 기관고유사업 최종보고서
[9] 전산원(1999), 국가주요전자도서관연계사업, 정보통신부 과제 최종보고서
[10] 행정자치부(1998), 전자정부의 비전과 전략, 행정자치부 자료
[11] A. Moffat, Economical inversion of large text files, Computing Systems, Vol. 5, No. 2, pp. 125-139, Apr. 1992.
[12] A. Moffat and J. Zobel, Coding for compression in full-text retrieval systems, Proc. 2nd IEEE Data Compression Conference, pp. 72-81, Mar. 1992.
[13] I. H. Witten, A. Moffat and T. C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, Van Nostrand Reinhold, New York, 1994
[14] T. C. Bell, A. Moffat, C. G. Nevill-Manning, I. H. Witten and J. Zobel, Data compression in full-text retrieval systems, J. of the American Society for Information Science, Vol. 44, No. 9, pp. 508-531, Oct. 1993.
[15] T. C. Bell, A. Moffat, I. H. Witten and J. Zobel, The MG retrieval system: compressing for space and speed, CACM, Vol. 38, No. 4, pp. 41-42, Apr. 1995.

강 무 영

1990 광주대학교 전자계산학과 졸업 (공학사)
1992 한남대학교 컴퓨터공학과 졸업 (공학석사)
1993~현재, 한남대학교 컴퓨터공학과 (박사과정)
1986~현재, 연구개발정보센터 연구원
관심분야: 데이터베이스, 검색엔진, 퍼지시스템
E-Mail: kmy@ns.kordic.re.kr

이 상 구

1978 서울대학교 전자공학과 졸업 (공학사)
1981 한국과학기술원 전산학과 졸업 (이학석사)
와세다대학 전기전자컴퓨터공학과 졸업 (Ph. D)
1983~현재, 한남대학교 컴퓨터공학과 교수
관심분야: 컴퓨터구조, 병렬처리, 퍼지이론, 데이터베이스
E-Mail: sglee@eve.hannam.ac.kr