

# 분산 정보 관리 시스템을 위한 사용자 API 설계 및 구현

주원균<sup>0</sup> 이민호 강무영  
한국과학기술정보연구원 정보시스템연구실  
(joo<sup>0</sup>, cokeman, kmy)<sup>0</sup>@kisti.re.kr

User API for a distributed information management system.

Won-Kyun Joo<sup>0</sup> Min-Ho Lee Moo-Young Kang  
Dept. of Information System, Korea Institute of Science and Technology Information

## 요 약

본 논문에서는 트랜잭션 기반의 데이터 관리와 분산 검색 기능을 갖춘 분산 정보 관리 시스템인 KRISTAL-2000의 기능을 사용자에게 제공하기 위한 1) 사용자 프로그램 인터페이스 와 2) 사용자 화면 인터페이스의 설계 및 구현을 목표로 한다.

## 1. 서 론

정보 시스템에서 다루는 전문 데이터의 수와 용량이 증가함에 따라 정보 시스템의 각 분야에서는 대용량 데이터의 처리에 초점을 맞춘 연구들이 많이 수행되었는데, 그 중 정보의 저장 및 검색 부분에서는 많은 발전이 있었다.

이런 과정에서 전체 데이터를 여러 개의 DB로 나누어, 한 시스템 혹은 여러 시스템에 분산 저장하고 이를 통합 검색하는 연구들이 수행되었다. 이런 일련의 연구들 중 하비스트(Harvest)[1]는 초기에 제안되었던 시스템으로, 간단하지만 분산 검색 시스템의 기본적인 구조를 제시하고 이를 직접 설계/구현하여 실패를 보인 시스템으로 유명하다. 또한 에이전트 기반의 분산 검색 모델을 선보인 Harness, 웹을 기반으로 하여 통합 검색 방법을 제시한 여러 메타 검색 엔진들, 프로토콜 기반으로 정보의 융합 검색 방법을 제시한 START[2] 등 많은 방법들이 시도되었고, 몇몇 시스템들은 여전히 개발되고 있다. 이런 연구들은 현재 분산 검색, 통합 검색, 분산 통합 검색이라는 용어들로 대표되고 있다.

본 논문에서 대상으로 한 KRISTAL-2000 시스템은 분산 검색의 맥락에서, 대용량 데이터의 분산 검색 기능을 기본으로 제공한다. 또한 그 동안 검색 시스템의 문제점으로 지적되었던 데이터 관리기능을 개선하기 위해, 데이터의 삽입, 수정, 삭제 등의 기능 수행 시 트랜잭션 처리와 회복 기법을 도입하여 제공함으로써, 보다 신뢰성 있는 데이터 관리 기능을 제공한다.

본 논문에서는 분산 정보 관리 시스템 KRISTAL-2000의 기능을 사용자에게 제공하기 위한 사용자 API의 설계와 그 구현에 초점을 맞추었다. 2장에서는 KRISTAL-2000 시스템에 대한 이해를 돕기 위해 시스템의 구조와 각 구성 요소 별 기능에 대해서 설명하고, 3장에서는 KRISTAL-2000 시스템의 서비스 별 연동 관계와 사용자 응용프로그램 인터페이스(API), 그리고 각 서비스 별로 기본적으로 제공하는 사용자 인터페이스에 대해 설명하고 4장에서 결론을 맺는다.

## 2. KRISTAL-2000 시스템 구조

분산 환경에서 정보의 원활한 관리와 효과적인 검색을 지원하기 위하여 KRISTAL-2000 시스템은 그림 1과 같은 구조로 설계/구현되었다. 시스템 구조에서 두드러진 특징은 주요 구성요소(JOB Scheduler, Fire, Set Manager, Data Manager)들이 독립적인 데몬 프로세스의 형태를 취하고 있어 별도의 시스템에 이식 가능한 형태로 설계되었다는 점이다.

시스템 구조의 이해를 돕기 위해서 각 구성 요소에 대해 간단히 설명한다.

- 작업 스케줄러(Job Scheduler) : 사용자와 밀접한 연관이 있는 부분으로서 사용자의 요청(온라인 문서 관리/검색)을 받아들여 각 서버 시스템에 전달하여 결과를 얻을 수 있도록 도와 주는 부분으로서, 데이터 관리기/검색기/셋 관리기와 상호작용 하에 작업을 수행한다.
- 데이터 관리기(Data Manager) : 온라인 문서 관리를 주요 기능으로 하는 요소로서 데이터 삽입/삭제/수정의 역할을 수행한다. 기능 수행 시 트랜잭션 처리를 수반한다.
- 검색기(Fire) : 검색을 담당하는 부분으로서, 불리언과 벡터 모델의 확장된 검색 방법을 제공하고, 셋 관리기와 상호작용을 함으로써 효율적인 검색을 제공한다. 또한 검색을 위한 포스팅 정보의 효과적인 수집을 위해서 Shore[3] 쓰레드를 사용하여 DB에 연결된다.
- 셋 관리기(Set Manager) : 검색 결과의 저장 및 관리를 담당하는 부분으로서, 사용자가 로컬에 검색 결과를 저장하지 않고서도 자신의 검색 결과를 탐색 및 관리 할 수 있도록 하고, 캐쉬(cache) 기능을 제공함으로써 검색기의 빠른 검색 수행을 보조한다.

각 구성 요소들 간의 중요 연결 관계는 그림 1에서 세 가지 선을 이용하여 설명하고 있는데, 각 관계는 동일 프로세스에 존재하거나 파이프(pipe)로 연결된 프로세스 혹은 소켓(socket)으로 연결된 프로세스 형태를 취하고 있다. 이러한 연결 관계는 각 구성 요소의 독립성을 보장한다.

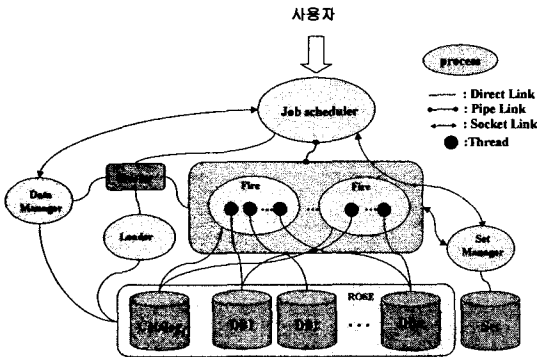


그림 1 KRISTAL 2000 시스템 구조

따라서 사용자 API는 이러한 시스템 구조 및 특성들을 반영함으로써, 사용자가 시스템의 모든 기능을 편리하고 쉽게 사용할 수 있도록 설계하여 구현되어야 한다.

그림에서 최상위에 사용자가 연결되는데, 분산환경임을 고려하여 사용자와 작업 스케줄러는 프로토콜을 이용하여 네트워크로 연결된다.

### 3. 사용자 인터페이스

3장에서는 정보 관리 시스템이 사용자에게 제공하는 2가지의 주요 서비스(온라인 문서 관리/검색)를 구분하여, 각 서비스 별 구성요소들의 연동 관계 및 응용프로그램 인터페이스(API)에 대해 설명한다

### 3.1 서비스 별 구성 요소들의 연동관계

#### - 온라인 문서 관리 서비스

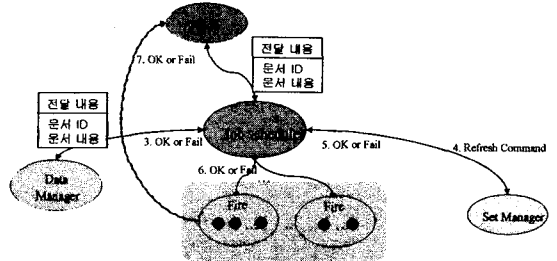


그림 2. 문서 관리 서비스 구성 요소의 연동 관계

사용자의 요청은 <문서 ID, 문서내용>을 입력으로 하여 작업 스케줄러에게 전달되고, 작업스케줄러는 데이터 관리기에 <삽입, 수정, 삭제>의 명령을 전달하여 올바른 데이터 관리 기능을 수행하도록 한다. 그 후, 셋 관리기의 결과 셋에 데이터 관리기의 변경 내용을 반영하도록 한다. 최종적으로 검색기가 결과내용을 사용자에게 전달한다.

#### - 검색 서비스

검색 서비스에서 다루는 내용은 DB에 대한 각종 정보들, 질의 검색 수행, 결과 셋 정보 및 문서 내용 정보의 획득에 관련된 것들이다.

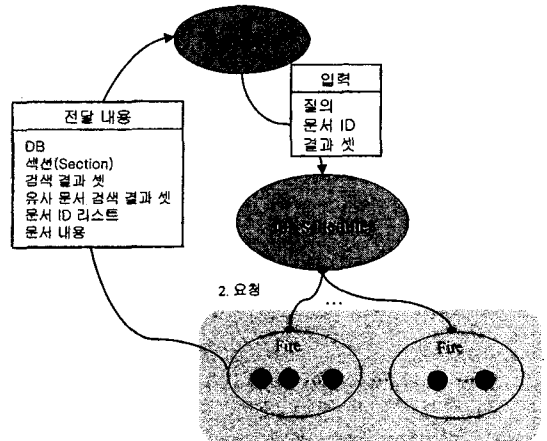


그림 3 검색 서비스 구성 요소의 연동 관계

사용자의 요청은 작업 스케줄러에게 전달되고 작업 스케줄러가 검색기를 이용하여 작업을 분배하여 결과를 사용자에게 전달한다.

### 3.2 사용자 API 명세

사용자 API는 3.1의 구성요소들의 연동관계에 기반하여 도출된 것으로 그림의 사용자라고 표기된 부분 바로 앞단에 위치하여 네트워크 프로토콜을 통해 사용자와 시스템을 연결시킨다. 여기에서 사용자라 함은 사용자에게 서비스를 제공하기 위한 일련의 프로그램들(명령어 방식, 웹 기반, 통합 검색 프로그램)을 지칭한다.

정보 관리시스템을 위한 사용자 API는 3 그룹으로 구성되는데, 각 그룹별 사용자 API 이름과 요약설명은 아래에서 설명한다.

- 그룹 1: 에러 처리 관련 API

CL_GetErrMsg	에러 코드와 메시지 출력
--------------	---------------

- 그룹 2: 온라인 문서관리 문서 관리 API

CL_GetPrimarykey	기본키 정보 구함
CL_AppendParsedDoc	파싱 된 문서 삽입
CL_UpdateParsedDoc	파싱 된 문서 갱신
CL_AppendUnParsedDoc	원본 문서 삽입
CL_UpdateUnParsedDoc	원본 문서 갱신
CL_DeleteDoc	문서 삭제

- 그룹 3: 검색 관련 API

CL_Search	확장된 불리언/벡터 검색
CL_SimSearch	유사 문서 검색
CL_GetDoclist	검색 결과 문서 리스트 구함
CL_GetContent	문서 내용 구함
CL_GetDBList	DB 리스트 구함
CL_GetSectionList	섹션 리스트 정보 구함
CL_GetMetaResult	검색 결과에 대한 정보 구함

### 3.3 화면 인터페이스 예제

그림 4는 차례대로 검색/간략보기/상세보기, 유사 문서 검색, 문서 삽입, 문서 수정/삭제관련 인터페이스 화면으로서 위에서 설명한 API를 이용하여 구현되었다.

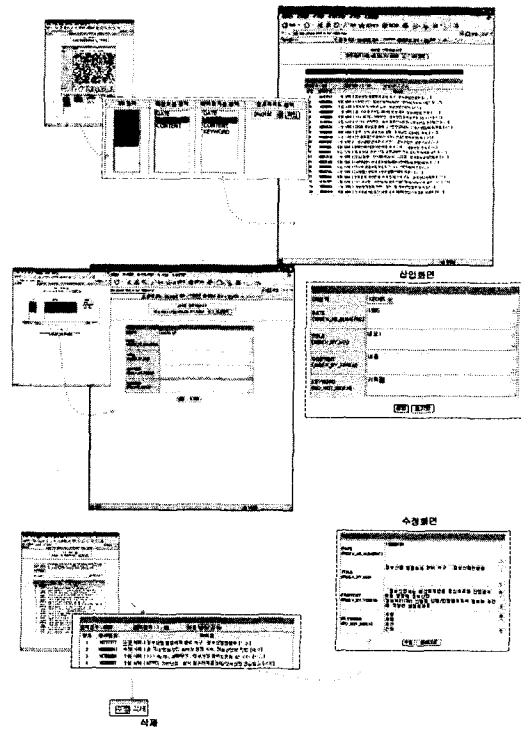
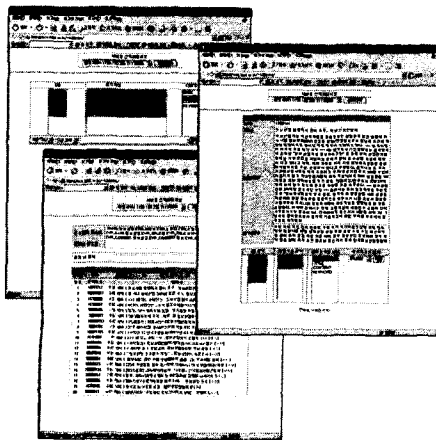


그림 4 화면 인터페이스 예제

### 4. 결론 및 향후 연구 과제

본 논문은 분산 정보 관리시스템을 이용하기 위한 사용자 API와 화면 인터페이스의 구현에 그 목적이 있다. 두 가지의 인터페이스가 설계/구현 되었는데, 먼저 사용자 API는 네트워크 프로토콜을 사용한 모듈을 통해 사용자와 시스템을 연결시키는 역할을 하고, 사용자 화면 인터페이스는 사용자들의 API 사용을 돕는 역할을 한다. 각 인터페이스는 C/C++로 구현되었다.

본 시스템에서는 설명되지 않았지만, 프로토콜에 관한 것은 KRISTAL 기술 매뉴얼[4]에 설명되어 있다.

#### [참고문헌]

- [1] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. *Harvest: A Scalable, Customizable Discovery and Access System* Technical Report CU-CS-732-94, August 26, 1994 Department of Computer Science, University of Colorado - Boulder.
- [2] Luis Gravano, Chen-Chuan K.Chang, Hector Garcia-Molina and Andreas Paepcke, "START: Stanford Proposal for Internet Meta-Searching", ACM SIGMOD Record, Vol.26, No.2, Pages 207-218, 1997
- [3] <http://www.cs.wisc.edu/shore/>
- [4] KRISTAL 기술 매뉴얼