제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

정보검색관리시스템을 위한 XML 기반 프로토콜 설계

이민호, 정창후, 주원균, 서정현, 류범종
한국과학기술정보연구원 정보시스템개발실
e-mail : cokeman@kisti.re.kr

A Design of XML based Protocol for Information Retrieval & Management System

Min-Ho Lee, C.H. Jeong, Wonkyun Joo, Jerry Hyeon Seo, Beom-jong You
Dept. of Information System Development
KISTI(Korea Institute of Science and Technology Information)

요 약

본 논문에서는 정형/비정형/XML 데이터의 검색 및 관리 기능을 갖춘 정보검색 관리시스템을 위한 서버와 클라이언트간의 통신 프로토콜을 설계하는 방법에 대해서 설명한다. 제시된 프로토콜은 정보검색관리시스템에 적절하게 설계되었으며, 확장성 및 호환성을 높이기 위하여 XML 메시지로 이루어져 있다. 그리고 통신 처리 효율을 증가시키기 위하여 TCP/IP 기반의 소켓 통신을 한다. 또한 본 논문에서는 이러한 프로토콜을 기반으로 정보검색관리시스템에서 기본적으로 제공하여야 하는 서비스를 정의한다.

1. 서론

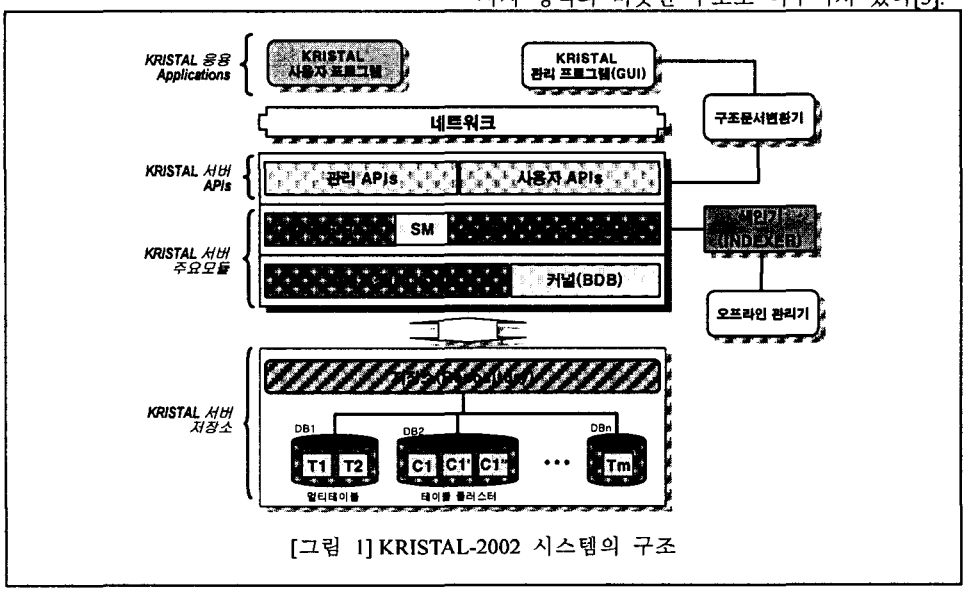
정보 시스템에서 다루는 전문 데이터의 수와 용량이 증가함에 따라 정보 시스템의 각 분야에서는 대용량 데이터의 처리에 초점을 맞춘 연구들이 많이 수행되었는데, 그 중 정보의 저장 및 검색 부분에서는 많은 발전이 있었다.

이런 과정에서 대용량 정보의 검색과 정보의 저장을 동시에 지원하기 여러 형태의 시도가 진행되어 왔다. 첫째로, 상용 DBMS 를 이용하여 정보에 대한 관리를 수행하고, 사용자 검색을 수행하기 위해서는 정보검색시스템을 사용하는 DBMS 과 정보검색 시스템의 연동 방식이 있다. 하지만, 이러한 환경은 DBMS 와 정보검색시스템을 이중으로 운영하여야 하는 불편함이 있고, 변동된 문서에 대한 DBMS 시스템에 저장된 문서와 정보검색 시스템에 저장된 색인정보와의 일관성을 유지하기 어렵고, 데이터를 중복 저장하므로 저장공간의 낭비를 가져오는 문제점이 있었다. 이러한 문제를 극복하기 위해 두 번째 방법인 정보관리 시스템과 검색 시스템이 밀 결합한 형태의 연구 등이 나타나기 시작했다. 국내의 경우를 예로 들면 바다-3 와 오디세우스[1] 등이 있으며, 최근에는 Google 이 관리기능은 없지만 저장 측면에서 보다 발전된 형태인 분산 저장 검색 시스템 GFS(Google File System)을 선보였다[2].

한국과학기술정보연구원(KISTI)에서는 정보검색시스템과 관리시스템의 밀 결합한 형태의 한 연구로서 정보검색관리시스템 KRISTAL-2002 를 개발하였다. KRISTAL-2002 는 정보검색시스템의 기술을 기반으로 DBMS 의 가장 기본적인 기능인 데이터의 추가, 삭제, 변경 등의 데이터 처리, 트랜잭션과 데이터 복구 등의 요소기술 등을 추가하여 기존 정보검색 시스템에 비해 안정적인 데이터 관리를 가능하게 하면서도 검색 성능을 떨어뜨리지 않는다[3]. 정보검색 관리시스템은 기존의 DBMS 가 가지는 강력한 관리기능을 점차적으로 수용하는 방향으로 계속 발전되어 가고 있다.

본 논문에서는 이러한 정보검색관리시스템의 개

발시 확장성과 호환성, 효율성 등을 고려한 프로토콜 설계를 목표로 한다. 또한 정보검색 관리시스템이 제공해야 할 기본적인 기능을 처리할 서비스를 정의한다. 2 장에서는 정보검색관리시스템인 KRISTAL-2002 의 구조 및 기능들에 대해서 간략히 설명하고, 3 장에서는 XML 메시지를 기반으로 하는 정보검색관리시스템을 위한 프로토콜을 설계한다. 4 장에서는 정보검색 관리시스템이 가져야 할 기본적인 서비스를 정의한 후, 5 장에서 결론을 맺는다.



[그림 1] KRISTAL-2002 시스템의 구조

2. KRISTAL-2002 의 구조 및 기능

KRISTAL-2002 는 데이터의 안정적인 관리 및 검색 속도를 보장하기 위하여 다음과 같은 구조로 설계되었다.(그림 1)

- 안정적인 하부저장 엔진

트랜잭션과 대용량 처리에 모두 효율적인 Berkeley DB[4]를 하부 저장 엔진으로 채택하여 안정성과 효용성을 높였다.

- 서버 풀 방식의 프로세스 구조

동시 사용자의 처리를 위하여 서버 풀 방식의 구조를 갖으며 로드 밸런싱을 담당하는 프로세스가 이를 관리한다.

- 빠른 검색 속도

캐시 기능을 하는 셋 관리기(SM)가 따로 있으며 검색은 다중 쓰레드 방식을 취해 빠른 검색 속도를 보장한다.

- XML 문서의 지원

XML 문서의 저장 및 검색을 지원하기 위해서 구조문서변환기 모듈이 있으며, 원본 XML 문서를 KRISTAL 에서 처리할 수 있는 중간 XML 문서로 변환하여 KRISTAL 로의 적재 및 서비스를 가능하게 한다.

- BLOB 데이터의 지원

BLOB 을 지원하기 위한 데이터 타입을 제공한다.

3. 정보검색관리시스템 프로토콜 설계

정보검색관리시스템 프로토콜은 XML 의 장점인 확장성과 가독성을 살려 XML 형태의 메시징 프로토콜로 설계되었으며, 이기종 시스템간의 상호 호환적인 메시지 전달을 위한 구조를 정의한 "OASIS 8.0"의 메시지 형식과 비슷한 구조로 이루어져 있다[5].

그림 2 에서 보듯이 설계된 프로토콜에서 XML 메시지는 크게 프로토콜에 관한 정보를 담고 있는 Header 부분과 각 서비스 파라미터와 관련된 Body 부분으로 구성되어 있다.

3.1 Header 의 내용

- Version

프로토콜의 버전(version)을 표시한다. 만약 프로토콜 버전이 맞지 않을 경우에는 버전 불일치 에러로 처리한다. 버전의 변경은 프로토콜의 구조가 변하는 Major 변환과 서비스명이나 파라미터가 변경되는 Minor 변환이 있을 수 있다. 버전의 표시는 "version 1.0" 과 같이 "Major 번호.Minor 번호"로 표기한다

현재는 Version 정보밖에 들어가 있지 않으나 향후 보안을 위한 정보, 통신속도를 향상시키기 위한 압축 정보 등이 들어갈 예정이다.

3.2 Body 의 내용

Body 는 서비스와 관련된 내용이 들어가며, Process 와 Object 로 이루어져있다. 즉, 'Object 를 가지고 Process 를 하라' 라는 의미이다.

```
<Message>
  <Header>
    <Version>
      1.0
    </Version>
  </Header>
  <Body>
    <Process>
      Search
    </Process>
    <Object>
      Parameter
      Parameter
      ...
    </Object>
  <Body>
</Message>
```

[그림 2] XML 메시징 프로토콜 구조

▪ 프로세스(Process)

서비스명을 나타낸다. 요청 메시지의 경우 영문자로 된 서비스명을 사용하고, 서버로부터 요청에 대한 응답 메시지인 경우에는 요청 메시지의 영문자 서비스명 뒤에 "_RESPONSE "를 붙인다. 예를 들어 요청 서비스가 검색일 경우, 요청 서비스명은 "RETRIEVE" 이고, 응답 서비스명은 " RETRIEVE_RESPONSE " 이다. 에러가 발생할 경우에는 요청 서비스명에 " _RESPONSE "를 붙이는 것이 아니라 에러를 나타내는 " ERROR " 을 사용한다.

▪ 객체(Object)

서비스에 관계된 파라미터를 나타낸다. 요청 메시지인 경우에는 입력 파라미터를 나타내며, 응답메시지인 경우에는 서버로부터의 요청 결과 값을 나타낸다. 파라미터 값들은 구조적인 형태를 가지고 각 서비스 별로 다르게 표현되는데, 이 값들은 구현되는 정보검색관리시스템에 종속적인 값들을 갖는다[6].

객체에서 사용하는 파라미터 값 중 문자열 타입의 파라미터는 Base64 코드를 사용하여 인코딩(encoding) 된 메시지를 전송하고, 메시지 수신부에서는 다시 원래의 문자열로 복원하기 위해서 Base64 디코딩(decoding)을 수행한다. 이를 통해 프로토콜에서 사용하는 문자나 정보검색관리시스템이 정의하는 특수한 문자로 인해 발생하는 Meta-Character 문제를 방지할 수 있다.

4. 정보검색관리시스템 서비스 정의

본 장에서는 정보검색관리시스템이 가져야 할 기본적인 서비스를 정의한다. 우선 다음과 같이 5 개의 그룹으로 나누었다.

- 검색 서비스

검색, 유사문서 검색, 결과내 검색등 검색에 관계된 서비스등이다.

- 서버관리 서비스

정보검색관리시스템의 종료, 상태 점검, 서버의 형태 등 서버자체에 대한 관리 서비스이다.

- 표현 서비스

검색된 결과 혹은 저장된 정보를 가져오는 서비스이다.

- 결과 변환 서비스

검색된 결과 집합 혹은 저장된 정보를 사용자가 원하는 임의의 형태로 변환하는 서비스이다.

- 데이터관리 서비스

데이터의 삽입, 삭제, 변경등 데이터 자체의 온라인 관리 서비스이다.

위와 같이 개념상의 그룹으로 나누어 정의하며, 각 그룹당 세부적인 서비스 내용은 정보검색 관리시스템에 따라 달라질 수 있다. KRISTAL-2002 에서는 다음과 같은 세부 서비스를 정의하였다.

✓ 검색 서비스

RETRIEVE : 검색
RETRIEVE_SIMILAR_DOCUMENTS : 유사문서 검색
RETRIEVE_IN_RESULT : 결과 내 검색

✓ 서버관리 서비스

CHECK_STATUS : 서버 상태 점검
KILL_SERVER : 서버 종료

✓ 표현 서비스

GET_DB_INFO : DB 저장, 색인, 섹션 정보 제공
GET_SET_INFO : 결과 집합의 정보 제공
GET_DOCUMENTS_FROM_RESULT : 검색 결과 제공
GET_DOCUMENTS_WITH_IDS : 문서 ID 를 통한 문서 내용 제공
GET_DOCUMENTS_WITH_PRIMARY_KEY : 기본키를 통한 문서 내용 제공
GET_XML_NODES_FROM_RESULT : 검색 결과로부터 XML 문서의 노드 (엘리먼트)를 제공
GET_XML_NODES_WITH_IDS : 문서 ID 를 통한 XML 문서의 노드 제공
GET_XML_TREE : XML 문서를 TREE 형식으로 제공

✓ 결과 변환 서비스

SORT_BY_SECTION : 정보를 섹션 기준으로 정렬

✓ 데이터 관리 서비스

PROCESS_DATABSE_SCHEMA : 데이터베이스 스키마의 변경
APPEND_DOCUMENT : 문서의 삽입
DELETE_DOCUMENT : 문서의 삭제
UPDATE_DOCUMENT : 문서의 변경
APPEND_XML_NODE : XML 노드의 삽입
DELETE_XML_NODE : XML 노드의 삭제

UPDATE_XML_NODE : XML 노드의 변경
MOVE_XML_NODE : XML 노드의 이동

데이터 전체보기나 외부 키등을 이용한 검색등 위에서 제시한 것 외에 많은 서비스들이 KRISTAL-2002에는 정의하여 개발되었다.

5. 결론

본 논문에서는 정형/비정형/XML 데이터의 검색 및 관리 기능을 갖춘 정보검색 관리시스템을 위한 서버와 클라이언트간의 통신 프로토콜을 설계하였다. 설계된 프로토콜은 확장 및 호환성을 높이기 위하여 XML 형식으로 이루어져있으며, 문자열을 Base64 코딩 방식을 사용하여 Meta-Character 문제를 제거하였다. 또한, 통신 처리 효율을 높이기 위하여 TCP/IP 기반의 소켓 통신으로 처리 효율을 증가시켰다. 정보검색 관리시스템이 가져야 할 기본적인 기능을 그룹으로 나누어 정의하였으며 KRISTAL-2002 의 예를 들어 상세한 서비스를 설명하였다. 향후 정보검색관리시스템 프로토콜에 보안기능과 압축기능 등을 추가하여야 하며 세부 서비스를 자세히 정의할 필요가 있다.

참고문헌

[1] 오디세우스, http://odysseus.kaist.ac.kr/
[2] Google, http://www.google.com/
[3] KRISTAL-2002 기술 매뉴얼, http://giis.kisti.re.kr/
[4] Berkeley DB, http://www.sleepycat.com/
[5] "OASIS 8.0" http://www.openapplications.org/oagis
[6] KRISTAL-2002 사용자 매뉴얼, http://giis.kisti.re.kr/