

# P2P 환경에서의 객체 기반의 데이터 모델링

노재금\*, 구태완\*, 이성룡\*, 정연진\*\*, 이광모\*

\*한림대학교 컴퓨터공학과

\*\*춘천정보대학 멀티미디어과

e-mail: jgrho@hallym.ac.kr

## An Object-based Data Modeling on P2P Environments

Jae-Gum Rho\*, Tae-Wan Gu\*, Sung-Ryong Lee\*

Yeon-Jin Jung\*\*, Kwang-Mo Lee\*

\*Dept. of Computer Engineering, Hallym University

\*\*Dept. of Multimedia, Chuncheon College of Information Technology

### 요 약

P2P(Peer to Peer)개념으로는 순수 P2P와 변형 P2P 구조가 있다. 순수 P2P 구조는 정보공유에서 연결성을 장점으로 하는 Gnutella의 형태를 응용한 형태가 많이 사용되고 있지만 정보의 검색 및 제공에 있어 수많은 트래픽이 발생하게 된다. 이와 달리 변형 P2P 구조들 중 서버의 역할을 최소화한 인덱스 서버 모델이 존재하는데 이 모델은 사용자에게 효율적이고 빠른 검색과 색인을 제공하기 위해 기존의 서버/클라이언트 형태를 취하고 있다.

본 논문에서는 질의 기법의 보편적인 방법인 JDBC(Java Database Connectivity)를 이용한 직접적인 질의 방법이 아닌 객체 지향 프로그래밍 환경에서 효율적인 질의 방법인 JDO(Java Data Object)를 이용하여 객체 기반의 데이터 모델링을 제안하며 이를 이용하여 인덱스 서버 개념을 적용한 파일 공유 시스템 (File Sharing System Using Data Modeling based on Object:FSUDMO)을 제안한다.

### 1. 서론

현재 초고속 정보통신망이 일반 가정에까지 보급됨에 따라 인터넷을 통한 정보 제공이 활발히 이루어지고 있다. 이러한 정보 제공을 위하여 최근 인터넷 비즈니스에서 P2P(Peer-to-Peer) 서비스가 등장하고 있다. 이는 기존의 서버 중심의 인터넷 환경에서는 클라이언트의 자원을 효율적으로 활용할 수 있는 방법이 부족하여[1], 클라이언트의 자원을 효과적으로 활용하기 위한 방법으로 P2P 기술이 나타나게 되었고, 냅스터(Napster), 그누텔라(Gnutella)와 같은 대표적인 P2P 어플리케이션으로 인하여 P2P 기술이 널리 퍼지는 계기가 되었다.

P2P 비즈니스 모델 탄생의 가장 중요한 하부적 배경은 첫째, 인터넷 대역폭의 광대역과 고속화. 둘

째, 인터넷 사용자들의 PC 성능의 고도화. 셋째, 서버 집중식 모델의 한계성과 비용이다.

본 논문에서는 효율적인 파일 공유 시스템의 설계를 위해 기존의 P2P 모델에서 인덱스 서버 모델을 분석한 뒤, 객체 지향 프로그래밍 환경에서 질의 방법의 편의성, 간결성 및 데이터 저장 형태의 독립성을 제공하는 JDO(Java Data Object)를 이용하고자 한다. JDO는 JAVA 문법 그대로를 이용할 수 있기 때문에 상속 개념에 근거한 데이터 모델링을 한 후 이를 접목하여 인덱스를 보다 효율적으로 관리하는 방법인 객체 기반의 파일공유 시스템을 제안한다.

2. 관련 연구

2.1 냅스터(Napster)

사용자가 서버에 접속한 후 자신이 가지고 있는 파일 리스트를 서버에 등록하면 서버는 파일 리스트를 데이터베이스에 저장한다. 사용자가 검색어를 서버로 보내면 서버는 검색을 한 후 검색 결과를 사용자에게 전송한다[2]. 직접적인 파일 전송만 개인 대 개인으로 이루어지게 된다. 이처럼 대부분의 작업(공유 및 검색)이 서버에서 이루어지게 때문에 사용자 증가에 비례한 서버 운용 비용이 많이 든다.

2.2 그누텔라(Gnutella)

사용자들의 컴퓨터가 중앙 서버를 통하지 않고 서로간에 직접 연결되어 있다. Gnutella 네트워크에 연결하고자 할 경우에는 연결되어 있는 1개 이상의 컴퓨터 주소를 입력한 후 그 컴퓨터들을 통해 Gnutella 네트워크에 연결한다. 사용자가 검색어를 입력하면 서로 연결되어 있는 컴퓨터들끼리 검색어를 포워딩한다[3]. 이와 같은 작동은 많은 양의 패킷(packet)이 피어(peer)들의 연결을 유지하는데 소모되기 때문에 불필요한 네트워크 자원이 소모된다. 또한 검색어가 포워딩되는 형태이기 때문에 데이터가 한 곳에 집중될 때에는 병목현상이 발생한다.

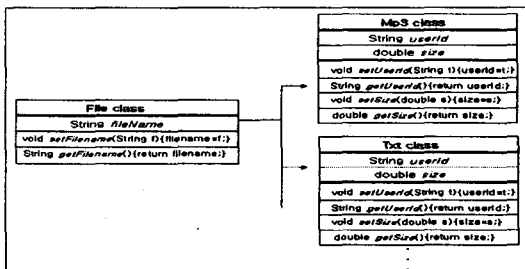
2.3 JDO(Java Data Object)

JDO는 자바 어플리케이션 객체와 데이터 저장소 사이의 단일화되고, 간단하고, 투명한 영속적인 인터페이스를 제공한다[4]. 객체 지향 프로그래밍이 보편화 됨에 따라 실행시간에 생성된 데이터에 대해 JDO를 이용하여 객체 형태로 저장이 가능해졌다.

3. 본론

3.1 데이터 모델링

데이터 모델링은 객체지향 프로그램 설계에 있어 첫번째 단계이다. 상속은 객체지향 언어의 장점인 모듈의 재사용과 코드의 간결성을 제공하는 중요한 특성이다. 본 논문에서는 클라이언트 프로그램으로부터 전송되어온 파일 목록을 정의된 데이터 모델링의 타입으로 각각 객체화하여 저장한다.

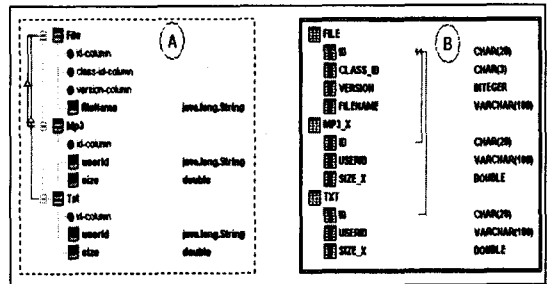


[그림 1] 상속 관계 예

“Mp3” 클래스와 “Txt” 클래스는 “File” 클래스로부터 상속되었다. 따라서 “Mp3” 클래스와 “Txt” 클래스는 “File” 클래스의 멤버변수인 fileName과 파일 이름을 저장하기 위한 함수인 setFilename()와 파일 이름을 반환하는 getFilename() 함수를 사용할 수 있다.

“Mp3” 클래스와 “Txt” 클래스는 각각 사용자 아이디를 저장하는 userID 멤버 변수와 파일 크기를 저장하는 size 멤버 변수 그리고 각 멤버 변수를 저장하고 반환하는 멤버함수로 이루어져 있다.

예를 들어, “Mp3” 객체가 생성되기 위해서는 “Mp3” 클래스의 상위 클래스인 “File” 클래스의 멤버변수인 fileName이 정의 되어야 한다.



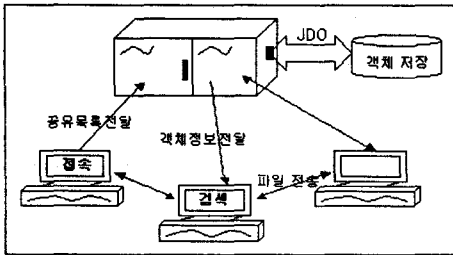
[그림 2] 데이터 모델링

위의 그림은 클래스 파일로 데이터 모델링한 결과로써, A부분은 각 객체가 가지는 속성과 클래스 간의 관계를 보여주는 것이고 B부분은 각 객체가 저장되기 위한 테이블 생성을 위한 것이다. 각 클래스에서 정의된 멤버 함수는 객체 정보를 할당하거나 얻기 위해서 사용된다. ID는 각 객체가 생성될 때 할당되는 것으로 관계형 데이터베이스에서 프라이머리 키(Primary Key) 역할을 하는 것으로 객체를 구별하기 위한 것이고, CLASS\_ID는 하위 클래스에서 최상위 클래스의 멤버 변수인 filename을 사용하기 때문에 하위 클래스를 구별하기 위한 것이다. 즉, “File” 객체에 같은 파일명을 가진 객체가 있을 경우, 그 파일명이 어떤 객체에서 상속되어 생성된 것인지 식별하여 준다.

예를 들어, test.mp3 파일과 test.txt 파일은 각각 “Mp3” 클래스의 객체와 “Txt” 클래스의 객체로 저장되고, “File” 클래스에 파일명이 test로 각각 저장된다. 이때 “File” 객체의 CLASS\_ID는 각각 Mp3와 Txt가 할당된다. “File” 객체의 CLASS\_ID 값과 같은 객체는 동일한 ID값을 갖는다.

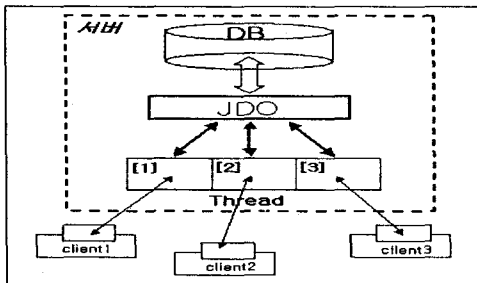
### 3.2 객체 기반 파일 공유 시스템 설계

객체 기반 파일 공유 시스템은 클라이언트로부터 파일목록을 받아 JDO를 이용하여 객체로 저장하고, 검색 요청이 오면 검색된 객체의 정보를 전송하게 된다. 클라이언트는 이 정보를 이용하여 다른 클라이언트에게 접속하여 파일을 다운로드하거나 자신에게 접속한 클라이언트에서 파일을 업로드하게 되는 메카니즘을 가지게 된다.



[그림 3] 객체 기반 파일 공유 시스템 구조

#### 3.2.1 인덱스 서버모델



[그림 4] 서버 구성도

서버는 클라이언트와 통신하는 부분과 파일 목록을 객체화하여 저장하는 부분으로 나눌 수 있다[5]. 각 쓰레드(thread)는 사용자 확인, 파일 목록 객체화, 검색 요청 처리를 담당한다.

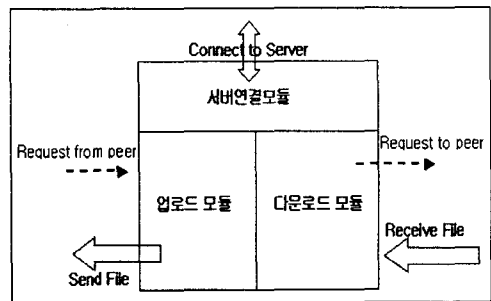
클라이언트가 접속을 요청하면 서버는 하나의 쓰레드를 할당한 후 올바른 사용자인지를 확인한다. 올바른 사용자가 아니면 연결을 끊고 다른 사용자가 이용할 수 있게 쓰레드를 회수한다[6]. 인증된 클라이언트는 클라이언트의 정보(사용자 ID, 클라이언트 IP, Port 번호)를 로그인사용자 테이블에 저장해서 현재 클라이언트가 서버에 접속해 있다는 것을 알 수 있게 한다.

인증된 클라이언트가 자신의 공유파일 목록을 서버로 전송하면 서버는 이전에 현재 사용자의 파일 목록에 대한 모든 객체들을 삭제한 후 파일 목록을 정의된 데이터 모델링의 타입으로 각각 객체화하여

저장한다.

클라이언트로부터 타입 구별없이 파일명만으로 검색 요청이 들어오면 서버는 최상위 클래스인 "File" 클래스의 객체들을 검색하고, 검색된 객체의 CLASS\_ID값에 같은 타입에서 "File"에서 검색된 객체의 ID 값과 같은 ID값을 갖는 객체를 찾는다. 타입과 함께 파일명으로 검색요청이 들어오면 타입과 같은 클래스("Mp3" 또는 "Txt")에서 검색한다. 클라이언트로 전송되는 정보는 검색된 객체의 정보(파일명, 파일크기)와 로그인사용자 테이블의 정보(IP, Port 번호)를 결합하여 전송한다.

#### 3.2.2 클라이언트 모델



[그림 5] 클라이언트 구성도

클라이언트는 서버와 연결을 유지하는 서버연결 모듈과 다른 클라이언트로부터 파일 전송 요청을 받아들이고 파일을 전송하는 업로드 모듈 그리고 자신이 다른 클라이언트에 파일 요청을 보낸 후 파일을 받는 다운로드 모듈부분으로 나눌 수 있다[5].

서버연결모듈은 사용자의 ID와 Password를 통해 서버에 연결 요청을 하고, 접속이 되면 자신의 공유 파일 목록을 서버로 전송한다. 이때 클라이언트가 파일 검색 요청을 하면 서버는 검색 결과를 보내게 된다. 업로드모듈은 자신의 공유 파일을 전송할 수 있는 하나의 포트를 열어두어 다른 클라이언트가 접속할 수 있게 한다. 업로드 포트에 접속한 다른 클라이언트가 요청한 파일이 없을 경우에는 연결을 끊고, 있는 경우에는 파일을 전송한다. 다운로드모듈은 서버연결모듈에서 서버로 검색 요청한 결과 목록에서 자신이 다운받기를 원하는 파일명을 선택하면 파일을 가지고 있는 클라이언트에 접속하여 파일을 요청을 하게 되고 파일 다운로드 받게 된다.

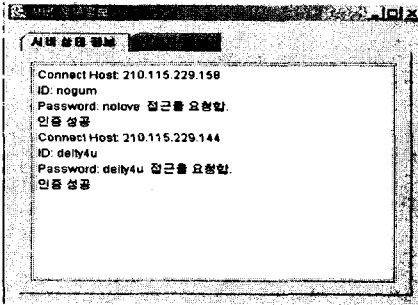
#### 3.2.3 활용분야

전자상거래시스템에서 품목별 또는 기능별로 구분

되어 있는 것을 상속 개념을 도입하여 Super-Sub 형태의 새로운 개념의 전자상거래시스템을 구축하거나 분산된 기업에서 각 부서의 데이터와 그 데이터를 처리할 수 있는 기능(메서드)을 객체화 하여 기업에서 서로의 객체를 이용하여 상호 업무를 협력하는 응용 분야에 활용이 가능하다고 본다.

5. 구현

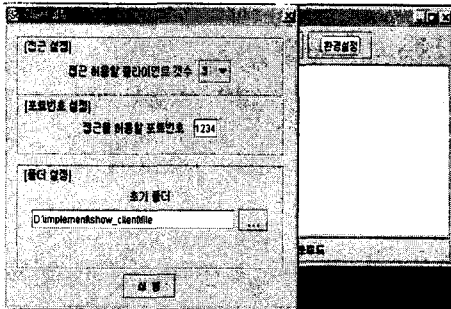
5.1 인덱스 서버



[그림 6] 서버 연결 상태

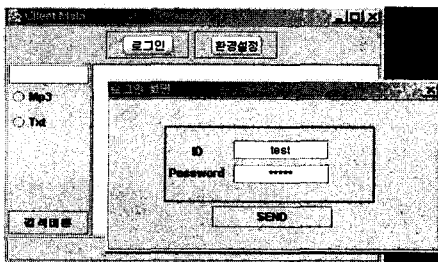
클라이언트 두 곳이 연결을 요청하여 인증을 받는 화면이다.

5.1 클라이언트



[그림 7] 클라이언트 환경 설정화면

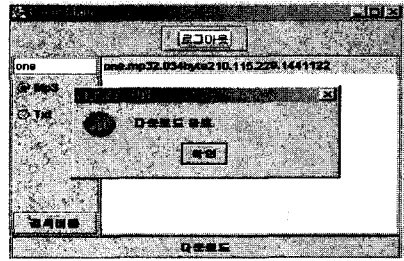
클라이언트는 공유 디렉토리와 접근을 허용할 클라이언트 수와 포트 번호를 설정한다.



[그림 8] 로그인 화면

ID와 Password를 이용하여 서버에 접속 요청을

보낸다.



[그림 9] 검색과 다운로드화면

자신이 검색하고자하는 객체의 타입을 설정하고 검색된 결과에서 하나를 선택하여 다운로드 받는다.

6. 결론 및 향후 연구 방향

본 논문에서 제안한 객체 기반 자료 공유 네트워크 시스템은 기존 P2P 방식의 인덱스 서버에서 JDBC를 통한 파일 목록의 저장과 관리를 사용하지 않았다. 논문에서 제안한 시스템은 JDO를 이용하여 상속 개념에 근거한 데이터 모델링을 통해 파일 목록을 객체화 하여 좀더 나은 인덱스 관리와 검색 방법을 제안하였다. 하지만 서버에서 특정한 서비스와 검색 등의 기능이 집중되어 이루어지고 있기 때문에 서버의 물리적, 논리적인 fail이 일어날 경우 서비스가 중단되는 단점이 있다. 향후 연구로 영속적인 서비스 제공에 대한 해결책과 개인 PC의 정보가 서버에 집중되어있기 때문에 서버 보안 정책이 필요하다

참고문헌

[1] G. Rief, E. Kirda, H. Gall, G.P. Picco, G. Cugola, P. Fenkan, "A Web-based peer-to-peer architecture for collaborative nomadic working", Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. WET ICE 2001. Proceedings. Tenth IEEE International Workshops on, pp.334-339, 2001

[2] <http://www.napster.com/>

[3] <http://gnutella.wego.com/>

[4] Java™ Data Object JSR 12 Version1.0

[5] 이광현, 외3명, "서버 부하 감소를 위한 P2P 기반 데이터 서비스 시스템의 설계 및 구현". 한국정보처리학회 논문지 C, VOL.9-C, NO.05, pp615~625, 2002, 10

[6] Allen Holub, 천영환 역, "자바 스프레드 능숙하게 다루기", 인포북, 2001. 5