

UML을 이용한 3-Tier 환경에서의 p2p 시스템 설계

(A P2P System Design Using UML Based On 3-Tier Environment)

이승아*, 권영직**
(Seung-A Lee, Young-Jik Kwon)

요약 인터넷의 발전은 사용자들의 정보에 대한 요구를 급속히 증가시키고 있으며, 고급 정보 획득을 위한 사용료에 대한 부담도 함께 증가하고 있다. 따라서 사용자 각자가 보유한 정보들을 다른 사용자들에게 오픈 시켜 공유할 수 있도록 하는 시스템의 개발에 대한 요구가 발생하게 되었다. 이러한 요구에 맞추어 본 논문에서는 네티즌들이 서로의 컴퓨터에 접속하여 상대방이 가진 정보들을 검색하고 공유할 수 있는 p2p(peer to peer) 시스템인 「p2p-3」을 제안하게 되었다. 「p2p-3」시스템은 객체지향설계 기법인 UML을 이용하여 개발함으로써 시스템 구현의 효율성을 추구하였다. 또한 일반적인 p2p 시스템의 문제점으로 대두된 보안문제를 해결하기 위하여 3-tier 분산시스템을 이용하였다.

1. 서론

인터넷의 발전은 사용자들의 정보에 대한 요구를 급속히 증가시키고 있으며, PC통신업체나 Portal, Content provider 등이 서비스하는 고급 정보를 얻기 위해 사용료를 지불해야만 했다. 그러나 이런 중개업체들을 거치게 되면 정보의 양이나 질적인 측면에서 한계가 있을 수 있고, 정보사용료에 대한 부담도 있을 수 있다. 따라서 사용자 각자가 보유한 정보들을 다른 사용자들에게 오픈 시켜 공유할 수 있도록 하는 시스템의 개발에 대한 요구가 발생하게 되었다. 이러한 요구에 맞추어 네티즌들이 서로의 컴퓨터에 들어가 정보들을 검색하고, 공유할 수 있는 시스템 즉, p2p(peer to peer) 시스템이 개발되게 되었다.

본 논문에서는 이러한 p2p 시스템을 객체지향설계 기법인 UML을 이용하여 개발함으로써 시스템 구현의 효율성을 추구하였다. 또한 p2p 시스템의 문제점으로 대두된 보안문제를 해결하기 위하여 3-tier 분산시스템을 이용하여 파일 공유 시스템인 「p2p-3」을 개발하였다.

2. 관련연구

(1) p2p 시스템

p2p 시스템은 Peer to Peer 즉, 중간자가 없는 동료간, 친구간의 E-Communication의 의미한다[3]. Forrester Research社는 “클라이언트들간의 상호작용을 조정하고 관리하는데 있어서 관리 서버 없이, 혹은 단일한 관리 서버를 통해서 클라이언트가 다른 클라이언트와 직접 연결할 수 있는 분산 컴퓨터 구조”를 p2p 기술로 정의하고 있다[4].

p2p 시스템은 개발 초기에는 mp3 형식의 음악 파일 등을 주고받는 개념에서 출발하였으며 1999년 미국의 손 패닝이 고안한 ‘Napster’가 그 시초라 할 수 있다[5,7]. p2p 시스템은 사용자 즉, 정보를 요구하는 클라이언트들의 관리를 위한 중앙 서버가 존재하는지 여부에 따라 하이브리드 p2p 방식과 순수 p2p 방식으로 구분할 수 있다[4].

(2) p2p 시스템의 분류

1) 하이브리드 p2p 방식(Hybrid p2p)

이 방식은 사용자들 즉, 클라이언트들 간의 로그인, 정보요구, 검색결과 제공, 내려받기 등의 작업을 관리해 주는 중앙 서버가 있어서 사용자가 새로 접

* 대구대학교 대학원 컴퓨터정보공학부 박사과정

** 대구대학교 정보통신공학부 교수

속할 때마다 목록을 갱신하며 관리한다[4]. 대표적으로는 1998년 1월 손 패닝이 개발한 Napster가 있다. 그러나 현재는 지적재산권 문제로 법원으로부터 서비스 중지 명령을 받아 파산하였다.

2) 순수 p2p 방식(Pure p2p)

미국 최대 종합 인터넷 업체인 AOL(America OnLine)의 자회사로 Winamp를 개발한 NullSoft에서 개발하여 공개한 'Gnutella'가 대표적이다[7]. Gnutella는 네트워크에 접속된 1개 이상의 컴퓨터 IP를 입력하고 Gnutella 네트워크에 접속한다.

국내에서도 서비스되고 있는 '소리바다'는 Napster와 Gnutella의 중간적 형태로 중앙 서버가 존재하지는 않지만 연결 상태만 관리하고, 연결된 사용자들의 파일을 검색하고 내려 받는 과정은 모두 사용자들 간에 일어난다[13].

p2p 시스템은 기술적인 문제도 가지고 있는데, 네티즌들의 접속이 폭주하면 연결이 끊어지거나, 진행속도가 느려지거나 검색을 원하는 정보들이 제대로 찾아지지 않는다. 또한, 보안성이 없기 때문에 사용자들의 각별한 주의가 요구되기도 한다.

(2)UML

UML이란 Unified Modeling Language의 약자로 객체지향 분석(Object-Oriented Analysis)과 설계(Design)를 위한 modeling language로 Rumbough, Booch, Jacobson의 방법론으로 통합하여 만들어졌으며 현재 ver1.1까지 만들어져 있다[8].

UML은 방법론(method)이 아닌 modeling language로 이해되는데, 대부분의 방법론들은 언어와 프로세스를 동시에 포함하는데 반해 modeling language는 그래픽 표현에 의해 단지 모델을 표현하는 것이라고 할 수 있다[10]. 따라서 UML을 사용하기 위한 개발 프로세스가 함께 정의되어야 한다. 또한 UML은 표준화된 언어로 시스템 개발과정에서 발생할 수 있는 의사소통의 불일치를 해결하기 쉽고 비교적 모순이 적은 다양한 표기법을 가진 언어이다[11]. UML은 시스템 전반에 대한 이해를 쉽게 할 수 있고, 프로젝트 규모의 변화에 유연하게 대처할 수 있으며, 배우기 쉽다는 이유 때문에 객체지향 응용기술에 있어서 널리 사용되고 있다[10].

UML은 기호로 사용되는 사물(Thing)과 이들 간의 관계(Relationship), 그리고 사물과 관계를 기반으로 그려지는 다이어그램으로 구성된다. 다이어그램은 기능에 따라 Use Case Diagram, Activity Diagram, Sequence Diagram, Collaboration Diagram, Class Diagram, Component Diagram, Deployment

Diagram으로 나뉘고 각 다이어그램은 서로 다른 관점에서 시스템을 표현한다[12].

(3) 3-Tier 분산시스템

3-Tier 분산시스템은 기존의 2-Tier 시스템의 문제점을 해결하기 위해 새로 도입된 방식으로, 사용자와 인터페이스 하기 위한 화면운동(presentation rule)을 클라이언트 시스템이 담당하고, 업무규칙(business rule)의 구현과 자료의 접근규칙(data access rule)의 운동을 서버시스템에서 담당하는 방법으로 서버와 클라이언트가 작업을 분담하는 형태로 시스템을 운용하는 방식이다. 이 방식에서 서버 시스템은 업무규칙을 처리하는 부분과 자료 관리를 위한 부분을 물리적으로 나누어 구성될 수도 있다[2].

이전 2-Tier에서는 네트워크 상에서 처리해야 할 통신량이 증가하고 클라이언트 수가 증가하면 성능이 떨어지고, 클라이언트가 분산되어 있어 응용프로그램 수정 시 모든 클라이언트에 각각 다시 수정해 주어야 하는 문제점을 가지고 있었다. 그러나 3-Tier에서는 클라이언트의 부하 및 자원의 증가로 인한 성능저하 문제를 해결하기 위하여 각 서버의 특징에 따라 애플리케이션 로직을 분산시켰다[1].

본 논문에서는 p2p 시스템 중 '접속 서버에 의한 p2p'를 구현하였다. 접속서버에 DB서버를 둔다는 것은 다수의 익명의 사용자들이 접속서버를 통해서 접속할 때 접속한 수만명의 데이터베이스 처리를 해야 한다는 의미이기 때문에 접속서버에서의 연결시켜주는 과정이 느려질 수 있고, 결과적으로 익명의 사용자들의 연결을 막는 이유가 될 수 있다.

또한 3-Tier 분산시스템을 통하여 접속서버와 데이터베이스서버를 뒀으로써 접속서버에서 보내는 질의문 외에는 접속을 통제하게 된다. 즉, MY-SQL의 포트번호는 3306이지만, 보통 FTP 나 HTTP의 포트는 21,80포트를 사용한다. 다시 말해서 3306 포트만 열어 놓고 다른 해킹가능 포트를 차단함으로써 보안 기능을 좀더 강화할 수 있다.

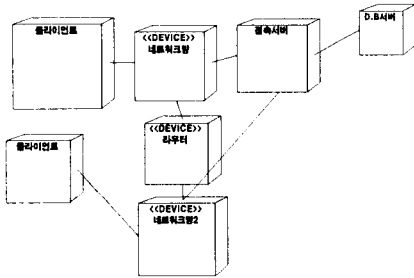
3. 구현된 시스템

본 논문에서 설계한 「p2p-3」는 3-Tier 분산시스템의 개념을 도입하여 종전의 p2p 시스템에서 취약했던 보안문제를 안전하게 처리하였고, Java로 구현하여 네티즌들이 사용하는 다양한 운영체제에 모두

사용 가능하게 하였다. 데이터베이스는 MY-SQL을 사용하였다.

또한, 객체지향 설계 기법인 UML을 이용하여 객체 단위로 시스템을 개발함으로써 시스템에 대한 새로운 요구나 유지·보수 등 환경변화에 잘 적응하도록 하였으며, 재사용성도 높였다. UML 모델링을 위하여 객체지향 분석/설계/구축 도구로 Rational Rose를 사용하였다. Rational Rose를 사용하면 설계 프로젝트의 복잡한 부분을 단순화시킬 수 있고, Use case를 만들어냄으로써 사용자의 관점에서 소프트웨어 시스템을 정의할 수 있다.

<그림 1>은 3-Tier 분산시스템을 이용한 p2p 배치 다이어그램을 Rational Rose로 표현한 것이고, <그림 2>는 Use case 다이어그램을 이용해서 시스템과 그 외부간의 상호 작용을 묘사하면서 사용자의 요구를 상세히 파악할 수 있게 하였다[12].



<그림 1> 3-Tier 분산시스템을 이용한 p2p 배치 다이어그램

<그림 2> USE CASE 다이어그램의 배경도

우선 Use case 사용자 요구분석을 살펴보면 다음과 같다.

1) 서버 <--> 클라이언트(1)

①프로그램 실행 시 공유 폴더가 지정되었는지를 확인한다.

①-1. 만약 없으면 공유폴더를 지정한다.

①-2. 있으면 통과한다.

②사용자가 접속한다.

③접속 서버에 접속한다.

④DB 서버에 접속 서버가 연결한다.

⑤IP와 Password를 체크한다.

⑤-1. 확인된 값이 정확하면 친구 목록을 보낸다.

⑤-2. 확인된 값이 틀리면 가입하라는 목록을 보낸다. 신규가입 한다.

⑥DB 서버에 접속서버가 연결되어져서 ID와 Password를 넣는다.

2) 서버 <--> 클라이언트(2)

⑦등록 완료됨을 보낸다.

⑧DB서버에서는 ID와 IP를 가져온다.

⑨클라이언트에서는 친구목록을 보여준다.

⑩클라이언트에서 찾고자 하는 정보의 키워드를 입력한다.

⑪클라이언트에서 연결서버에게 찾을 단어와 찾고자 하는 클라이언트 IP를 보낸다.

⑫연결서버에서는 연결된 IP를 DB서버에 보내주고 연결된 IP 값들의 클라이언트들에게 각각의 UDP를 클라이언트들에게 보내서 파일을 찾는다.

⑬클라이언트에서 친구 목록을 클릭한다.

3) 서버 <--> 클라이언트(3)

⑭클라이언트에 친구목록을 접속서버에서 받는다.

⑮접속서버에서는 친구목록 ID, IP를 받아서 IP를 UDP로 IP와 함께 친구 클라이언트로 보낸다.

⑮-1. 친구 클라이언트에서는 받은 IP로 공유 폴더내의 파일을 보낸다.

⑮-2. 자신 클라이언트에서는 받은 파일 내에 파일을 선택한다.

⑮-3. 친구 클라이언트와 연결해서 파일을 전송한다.

4) 클라이언트 <--> 상대 클라이언트

①자신 클라이언트는 파일을 찾는다.

②자신 클라이언트는 다른 클라이언트로부터 찾고자 하는 파일이 문자로 들어온다.

③자신 클라이언트는 다른 파일들의 리스트 중에서 하나를 선택한다.

④클릭 한 파일의 IP로 다른 클라이언트로 접속한다.

⑤파일전송을 실시한다.

⑥마친 후 파일 전송이 제대로 마쳐졌는지 이전에 전송된 문자와 길이가 같은지 확인한다.

5) DB서버 <--> 연결서버

①연결서버에서 ID, Password, 친구목록을 받는다.

②DB서버와 ID와 Password를 검색한다.

②-1. ID와 Password가 일치할 때에는 연결 성공을 알려준다.

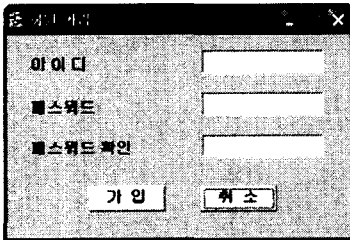
②-2. ID와 Password가 일치하지 않을 때에는 실패를 알려준다.

③ID와 IP를 초단위로 한번씩 udp에 보내줘서 DB에 클라이언트 접속여부를 확인한다.

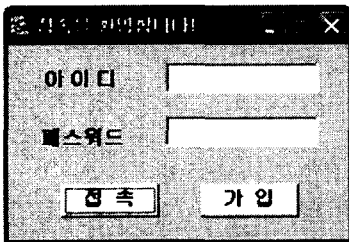
④②-1.이후 친구목록의 ID를 검색해서 연결서버로 보낸다.

⑤회원등록 시 DB서버에 ID와 Password를 저장한다.

<그림 3>은 회원가입 인터페이스이고, <그림 4>는 로그인 인터페이스이다. <그림 5>은 「p2p-3」의 메인화면으로 ‘찾을 파일이름’을 입력하고 ‘파일 찾기’버튼을 선택하면 접속되어 있는 클라이언트들이 공유한 폴더에서 입력한 키워드에 적합한 파일을 찾아준다. 내려받을 파일을 선택한 후 ‘내려받기’ 버튼을 클릭하면 된다.



<그림 3> 회원가입 인터페이스



<그림 4> 로그인 인터페이스

4. 실험 결과

1) 실험환경

본 논문에서 구현된 「p2p-3」 시스템은 다음의 다양한 환경에서 실험되었으며 연결서버와 데이터베이스 서버, 클라이언트 등의 역할에 사용된 시스템 사양은 아래 <표 1>과 같다.



<그림 5> p2p-3의 메인화면

<표 3> 실험환경

기능	사양
연결서버	CPU 600MHZ, RAM 128MB, HDD 15GB
MYSQL 서버	CPU 600MHZ, RAM 128MB, HDD 15GB
클라이언트1	CPU 500MHZ, RAM 128MB, HDD 15GB
클라이언트2	CPU 500MHZ, RAM 64MB, HDD 10GB
클라이언트3	CPU 500MHZ, RAM 64MB, HDD 10GB
클라이언트4	CPU 700MHZ, RAM 128MB, HDD 20GB

구현된 「p2p-3」 시스템을 실험한 네트워크 환경은 E1선로를 사용하였는데, E1선에서는 기본적으로 2Mbps정도의 전송속도를 가진다.

2) 실험결과

클라이언트에서 찾고자 하는 파일명을 입력하였을 때 각각의 클라이언트에서 미리 지정된 디렉토리에 파일명을 전송하여 일치하는 파일을 검색한다. 각 파일은 실험을 한 시스템들은 같은 허브 내에 있기 때문에 속도의 지장을 받지 않았다. 그러나 파일을 전송할 때 시스템 사양이 차이가 나면 속도가 일정하게 나오지 않는 문제점이 있었다. 예를 들어, CPU의 속도가 빠른 700MHZ 클라이언트에서 500MHZ 클라이언트로 파일을 전송할 때는 같은 700MHZ 클라이언트 간의 파일 전송 때보다 느리게 전송되었다. 이는 같은 허브를 통해서 파일이 전송되기 때문에 낮은 CPU처리 속도에서 처리되는 파일 전송속도를 따라 가는 것으로 해석할 수 있다. 그리고, 파일을 주고받는 각 클라이언트가 각 다른 허브에 위치할 때 라우터나 허브를 통과하는 속도 차에 의한 전송속도가 저하되는 현상도 발생하였다.

5. 결론 및 향후 연구방향

본 논문에서는 인터넷 사용의 급증과 함께 많은 사용자들이 관심을 가지고 사용하는 p2p 시스템을 UML을 이용하여 설계하고 구현하였다. UML 모델링을 효율적으로 지원하기 위해 Rational Rose를 이용함으로써 요구사항 분석 설계 단계에서 설계자가 전달하고자 하는 것을 보다 효과적으로 가시화 하여 시스템을 효율적으로 구현할 수 있었다.

「p2p-3」은 시스템 Java를 이용하여 클라이언트가 사용하는 운영체제에 상관없이 독립적으로 실행될 수 있도록 하였다. 「p2p-3」은 3-Tier 분산시스템 환경을 적용하여 클라이언트와 서버가 특징에 따라 역할을 분담하게 되므로 클라이언트의 접속이 폭증하더라도 파일을 내려받는 속도가 저하되지는 않았다. 또한 3-Tier 분산시스템 환경으로 구현되어 접속서버를 따로 관리하므로 p2p 시스템에서 문제점으로 대두되었던 보안과 DB 관리 문제도 함께 해결하였다.

그러나, 자바의 API를 Rational Rose의 기본 사양에 다 올려야 하기 때문에 과도한 메모리 낭비가 있고, 파라미터 전달값에 있어서 전달되는 값이 정확하지 않으면 코드생성 후 클래스간의 관계가 불분명해져서 프로그램 코드를 다시 만들어야 하는 문제점이 발생하였다. 이러한 문제점으로 인해 기존에 자신이 만든 클래스를 쓸 경우 클래스 다이어그램에서는 다시 정의하여야 하므로 이는 클래스 재사용 측면에서 고려되어야 할 부분이라고 할 수 있다.

접속하는 클라이언트의 수가 동시에 어느 정도일 때 시스템의 성능이 저하되는지, 클라이언트들이 각기 다른 허브에 접속되어 있을 때의 전송속도, 파일의 크기에 따른 전송속도의 측정 등도 추후 연구되어야 할 부분이다.

참 고 문 헌

- [1][http://denver.hyungwon.ac.kr/\[Appaz\]/메뉴얼/Visual%20Basic/VB_DataBase/lecture6.htm](http://denver.hyungwon.ac.kr/[Appaz]/메뉴얼/Visual%20Basic/VB_DataBase/lecture6.htm)
- [2]<http://gaston.snu.ac.kr/3tier>
- [3]이연주, "p2p의 현황과 진행방향", 테이콤, 2001.3.7
- [4]오세현, "p2p 방식의 파일 공유 서비스: 현황과 전망", 정보통신산업연구실, 2000.11.1
- [5]Napster, <http://www.napster.com>
- [6]Gnutella, <http://gnuella.wego.com>
- [7]이경전, "Napsterize Your Business", Enable 2000년 9월호.
- [8]윤청, 소프트웨어 공학, 생능출판사, 2000
- [9]G.Booch, I.Jacobson, and J.Rumbaugh, The Unified Modeling Language User Guide, The Addison-Wesley
- [10]왕창중, 소프트웨어 공학, 정익사, 1998.
- [11]T.Quatrani, Visual Modeling with Rational Rose and UML, Addison-Wesley, 1998.
- [12]조완수, UML 객체지향 분석 설계, 홍릉출판사, 2000.
- [13]소리바다, <http://www.soribada.com>