韓國 컴퓨터情報學會 論文誌
第13卷 第6號, 2008. 11.

2008-13-6-3-2

효과적인 협업을 위한 VNC 확장

이 태 호*, 이 홍 창**, 박 양 수***, 이 명 준***

Extending VNC for Effective Collaboration

Tae-Ho Lee *, Hong-Chang Lee **, Yang-Su Park ***, Myung-Joon Lee ***

요 약

*VNC*는 *Thin Client 컴퓨팅 시스템*의 하나로서 *RFB(Remote Frame Buffer) 프로토콜*을 통하여 서로 다른 플랫폼간의 원격 제어 기능을 지원한다. 그러나 VNC는 협업을 위하여 특별한 기능을 제공하지 않기 때문에 VNC의 원격 제어 기능을 통하여 효율적인 협업을 진행하는 것은 어려운 일이다. 본 논문에서는 오픈소스 VNC인 *UltraVNC* 서버와 *JavaViewer* 클라이언트의 확장에 대하여 기술한다. 효과적인 *실시간 협업*을 지원하기 위하여 확장된 VNC는 협업 서버를 제어하기 위하여 *관리자, 작업자, 관람자*와 같은 세가지 접근 권한을 제공한다. 관리자는 확장된 JavaViewer에서 새롭게 제공되는 *접근 관리 도구*를 통하여 권한 관리를 수행할 수 있다. 작업자는 협업서버의 자원을 제어하는 것이 가능하지만, 관람자는 협업서버의 자원을 제어 하는 것은 불가능하며 작업 진행사항을 단지 모니터링 할 수 있다. 또한 확장된 VNC는 여러 협업참여자가 간편하게 협업 서버에 접속할 수 있도록 *원클릭 접속 기능*을 제공한다.

Abstract

VNC is one of popular *thin client computing* systems, which supports cross-platform *remote control* using the *RFB* protocol. Unfortunately, since VNC does not provide functions specially designed for collaboration, it is difficult to collaborate effectively through the remote control function of VNC. In this paper, we present the extension of the *UltraVNC* server and the *JavaViewer* client which are one of open-source VNC systems. For effective *real-time collaboration*, the extended VNC provides three kinds of access authorities to control the collaboration server: *administrator, worker*, and *spectator*. The administrator can control the access authorities of the users through the access control tool, newly provided in the extended JavaViewer. The workers can access the resources of the remote server, whereas the spectator cannot handle those remote resources, only monitoring the remote server. In addition, the extended VNC provides *the one-click connection facility* for easy connection to the collaboration server from many collaboration participants.

▸ Keyword : VNC, Thin client Computing, RFB protocol, 원격 제어 (Remote Control,), UltraVNC, JavaViewer, 실시간 협업 (Real-time Collaboration), 관리자 (Administrator), 작업자 (Worker), 관람자(Spectator), 접근 관리 도구 (Access control tool), 원클릭 접속 기능(One-click connection facility)

• 제1저자 : 이태호 교신저자 : 이명준
• 접수일 : 2008. 8. 11, 심사일 : 2008. 9. 25, 심사완료일 : 2008. 11. 26.
* 울산대학교 컴퓨터정보통신공학부 석사과정 ** 울산대학교 컴퓨터정보통신공학부 박사과정
*** 울산대학교 컴퓨터정보통신공학부 교수
※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었습니다.(IITA-2008-(C1090-0801-0039))

Ⅰ. 서 론

Thin Client 컴퓨팅[1]은 서버 기반 컴퓨팅[2]의 한 종류이며, 서버의 화면을 출력하고 서버에 입력을 전송하는 기능만을 갖춘 간단한 클라이언트를 통하여 서버에서 모든 정보 관련 작업을 진행하는 컴퓨팅 개념이다. Thin Client 컴퓨팅은 언제 어디서나 정보에 접근하여 사용할 수 있으며 정보를 보호하는데 유리하다. 또한 Thin Client 컴퓨팅에서는 서버에 모든 소프트웨어를 설치하여 사용하므로 이를 사용하는 클라이언트에 대한 부담을 줄일 수 있고 서버에 대한 유지보수만 필요하므로 시스템을 안정성 있게 유지할 수 있는 등의 많은 장점이 있다[3].

VNC(Virtual Network Computing)는 Thin Client 컴퓨팅을 위한 원격 제어 소프트웨어이다[4]. VNC는 GUI(Graphic User Interface)를 통하여 여러 클라이언트가 동시에 다른 곳에 위치한 시스템의 자원을 사용할 수 있다. 또한 VNC는 단일 세션에 여러 클라이언트가 접속할 수 있는 환경을 제공한다. 이러한 VNC의 특징을 이용하여 분산되어 위치한 협업작업자들이 실시간 원격 협업을 진행할 수 있다.

협업 참여자가 VNC 클라이언트를 사용하여 VNC 서버에 접속하여 원격시스템의 자원을 사용하기 위해서는 VNC 서버의 사용자 인증 과정을 거쳐 신뢰할 수 있는 클라이언트임을 인증 받아야 한다. 인증을 위해 사용하는 수단은 협업 참여자가 VNC 서버에 접속할 때 서버에 미리 설정해둔 암호를 입력하는 것이다. 따라서 모든 협업 참여자가 원격시스템의 자원을 사용하기 위해 VNC 서버의 암호를 알아야 하므로 모든 협업 참여자에게 암호를 배포해야하는 문제점이 있다. 또한 한번 배포된 암호를 협업참여자가 아닌 다른 이가 소유하였을 경우 원격시스템의 보안이 허술해지는 문제점이 생기며, 그러한 문제점이 발생되면 암호를 다시 설정하고 협업참여자에게 변경된 암호를 배포해야하는 등의 암호 관리에 대한 문제점이 발생한다. 따라서 VNC를 이용하여 신뢰성 있는 협업을 수행하기 위해서는 위와 같은 문제점을 보완할 수 있는 방법이 요구된다.

본 논문에서는 실시간으로 이루어지는 협업작업을 지원할 수 있도록 확장된 UltraVNC 서버와 JavaViewer VNC 클라이언트에 대하여 기술한다. 확장된 VNC 클라이언트는 UltraVNC 서버 환경 설정 내용을 지원할 수 있도록 하였으며, 협업참여자가 한 번의 클릭으로 JavaViewer 클라이언트를 이용하여 UltraVNC 서버가 작동하는 협업서버에 접속할 수 있는 기능을 제공한다. 그리고 효율적인 협업작업 진행을 위해 VNC 서버에 접속한 협업참여자의 협업서버 자원 제어 여부를 관리할 수 있는 *관리자 권한*을 두었다. 그리고 협업서버의 자원을 제어할 수 있는 작업자 권한과 작업의 진행사항을 관람하는 것은 가능하나 협업서버의 자원을 제어할 수 없는 *관람자 권한*을 두었다. *관리자 권한*의 협업참여자는 확장된 VNC가 제공하는 권한 관리 기능을 통해 협업참여자에게 작업자 권한 또는 관람자 권한을 부여할 수 있고, 협업참여자가 아닌 협업서버 접속자를 강제퇴장 시킬 수 있다. 이와 같은 확장된 VNC가 제공하는 기능을 이용하여 효율적인 실시간 협업작업을 진행할 수 있다.

본 논문의 구성은 다음과 같다. 1절에 이어 2절에서는 확장 대상이 된 VNC 서버와 JavaViewer VNC 클라이언트 프로그램에 대한 소개를 한다. 3절에서는 협업서버에서의 권한 정의와 관리에 대한 설계와 구현을 기술한다. 4절에서는 간편한 협업서버 접속을 지원하는 원클릭 접속을 위한 환경에 대해 설명하고 5절에서는 개발한 시스템에 대한 평가를 내린다. 6절에서는 결론과 향후 연구 과제를 제시하고 있다.

Ⅱ. 관련연구

2.1. VNC의 소개

VNC는 Thin Client 시스템의 하나로써 서버와 클라이언트 프로그램으로 구성되며 클라이언트보다 서버에서 대부분의 작업이 이루어진다. VNC 서버는 윈도우즈와 X-window와 같은 GUI 플랫폼을 다른 PC의 VNC 클라이언트를 이용하여 제어할 수 있는 환경을 제공한다. VNC 서버와 클라이언트 프로그램은 TCP/IPv4 기반의 네트워크 연결을 통하여 RFB(Remote Frame Buffer) 프로토콜[5] 메시지를 주고받는다. RFB 프로토콜은 연결 초기화, 인증, 명령 전송, 결과 전송 등과 같은 VNC 서버와 클라이언트가 주고받는 메시지에 대하여 정의하고 있다. 현재 배포되고 있는 VNC로는 Real VNC, Tight VNC, UltraVNC가 있으며 모두 오픈소스 형태로 배포되고 JavaViewer 클라이언트와 윈도우즈 플랫폼을 원격제어할 수 있는 서버를 제공한다[6,7,8]. 또한 각 VNC는 부가기능을 구현하기 위해 RFB 프로토콜을 확장하여 사용하고 있다.

본 연구의 확장 대상이 되는 UltraVNC[8]는 윈도우즈 플랫폼을 원격제어하기 위하여 개발되었으며 윈도우즈 플랫폼의 특성에 맞추어진 원격 제어 기능을 제공한다.

UltraVNC가 제공하는 기능의 대표적인 것으로 MS-Logon, Mirror Driver가 있다[8]. MS-Logon 기능은 윈도우즈 운영체제의 사용자 명과 사용자 암호를 사용한 VNC 인증절차를 지원한다. Mirror Driver 기능은 클라이언트에 화면을 전송하기 위하여 발생하는 오버헤드를 줄이는 기능이다. UltraVNC의 서버 환경 설정 내용은 윈도우즈 레지스트리에 등록된다. 레지스트리에 등록된 서버 환경 설정 내용은 UltraVNC 서버를 실행할 때 별다른 입력 없이 서버 환경 설정 값을 읽어 들일 수 있어 편리하나, 하나의 환경 설정 내용만을 저장할 수밖에 없고, 서버 환경 설정 내용을 백업하기 위해서는 레지스트리 백업을 이용하여야 하며, 서버 환경 설정 내용을 보조기억 장치로부터 불러오기 위해서는 백업해 둔 레지스트리 내용을 가져오는 기능을 이용하여야 한다. 자바언어로 개발된 JavaViewer는 설치하여 사용하고자하는 플랫폼의 제한이 없으며 특히 애플릿으로도 동작하므로 사용자가 웹브라우저를 사용하여 VNC서버에 접속할 수 있는 기능을 지원한다. UltraVNC 서버와 함께 배포되는 JavaViewer는 네이티브 언어로 개발된 클라이언트와 동일한 기능을 지원한다. JavaViewer를 통하여 UltraVNC 서버에 접속하기 위해서는 실행 시 UltraVNC 서버의 IP 주소와 포트 번호를 입력하여 주어야 한다.

2.2. 클라이언트의 접속을 처리하는 UltraVNC 서버의 구조

UltraVNC 서버는 효과적인 원격 제어 기능을 제공하기 위하여 여러 가지 클래스로 이루어진 구조를 가진다. UltraVNC 서버의 클래스 가운데 클라이언트의 접속과 관계되는 주요 클래스는 다음과 같으며 그 관계를 [그림 1]에서 보이고 있다.

- vncserver : UltraVNC 서버를 실행하였을 때 발생하는 모든 처리를 담당한다. 또한 vncclient 클래스를 이용하여 클라이언트가 접속하였을 때 세션 객체를 생성하며 클라이언트의 접속을 관리한다.
- vncclient : 클라이언트에 대한 입력과 출력을 위하여 각 클라이언트와 UltraVNC 서버는 세션을 생성한다. vncclient 클래스는 서버와 클라이언트 간에 생성된 세션에서 필요한 기능을 제공하며 vncclient 객체는 서버와 클라이언트간의 세션객체이다. 클라이언트의 인증을 처리하며 JavaViewer와의 데이터 송수신을 담당한다.
- vncclientList : 세션 객체를 목록으로 관리할 수 있는 클래스이다. vncserver는 이 클래스의 객체인 m_unauthclients에 인증을 하지 않은 세션 객체를 모으며, 인증을 통과한 세션 객체를 m_authclients에 모은다.
- vncProperties : UltraVNC 서버의 설정을 담당하는 클래스이다. View Only와 같은 접속환경에 대한 설정과 암호의 입력과 저장을 담당한다.

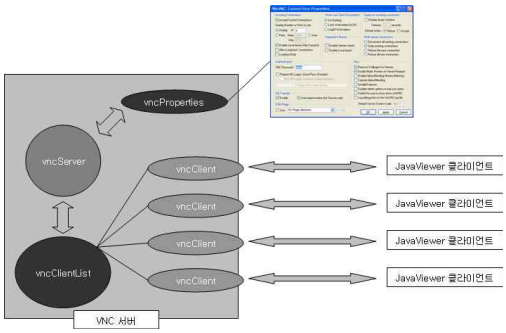

그림 1. UltraVNC 클래스간의 관계
Fig. 1. Class relationship on UltraVNC

2.3. RFB 프로토콜

RFB 프로토콜은 VNC 서버와 클라이언트의 통신에서 사용하는 메시지를 정의하고 있다. RFB 프로토콜을 이용한 통신은 ①Handshaking, ②Initialization, ③Server-to-Client, Client-to-Server Message 순으로 이루어진다[5]. 세 번째 과정에서 Server-to-Client 메시지는 VNC 서버의 화면과 클라이언트가 전송한 입력에 대한 처리결과를 클라이언트에 전송하는 메시지를 정의하고 있다. 그리고 Client-to-Server 메시지는 클라이언트가 서버에 전송하는 입력에 대한 메시지를 정의하고 있다[5].

III. 확장 VNC의 설계와 구현

3.1. 확장 VNC의 사용 권한과 관리

실시간 협업을 위하여 UltraVNC 서버와 JavaViewer를 사용하면 몇 가지 문제점이 발생한다. 협업서버 사용 권한을 정의하고 사용 권한을 관리할 수 있도록 UltraVNC 서버와 JavaViewer를 확장하여 드러난 문제점을 해결할 수 있다.

3.1.1. UltraVNC를 이용한 협업의 문제점

첫번째로 인증을 위한 암호의 배포와 관리에 관한 문제이다. 클라이언트가 UltraVNC 서버가 제공하는 원격시스템의 자원을 사용하기 위해서는 UltraVNC 서버에 미리 입력한 암호를 통한 인증과정을 마쳐 신뢰할 수 있는 클라이언트임을 인증받아야한다. 따라서 UltraVNC를 이용하여 협업을 진행하기 위해서 모든 협업참여자가 UltraVNC 서버에 설정된 암호를 알아야 하므로 암호 배포의 문제와 암호 관리의 문제가 발생한다.

두번째로 UltraVNC 서버에 접속하여 원격시스템의 자원을 제어할 수 있는 클라이언트의 권한이 일괄적으로 적용되며 관리가 어려운 문제이다. UltraVNC에는 클라이언트가 전송하는 입력을 무시하도록 하는 View Only 기능을 제공하고 있다. 그러나 이 기능은 모든 클라이언트에 일괄적으로 적용되어 특정 클라이언트를 선택하여 적용할 수 없다. 그리고 서버의 환경 설정을 통하여 적용되므로 View Only 기능을 사용하기 위하여 서버에 항상 관리자가 대기하여야 하는 문제점이 있다.

3.1.2. 협업서버 사용 권한 정의

UltraVNC를 통한 협업 작업에서 제어권한이 일괄적으로 적용되는 문제를 해결하기 위하여 다음과 같이 세 가지 협업서버 사용 권한을 정의하였으며 [표 1]에서 각 권한이 협업서버에서 가능한 작업을 보이고 있다.

- 관리자 권한 : 협업참여자의 접속 및 협업서버 사용 권한을 관리하고 협업서버의 환경을 관리하는 권한이다. 작업자와 관람자 권한의 협업참여자의 권한을 변경할 수 있으며 허용하지 않는 협업 참여자의 접속을 강제로 해제할 수 있다.
- 작업자 권한 : 협업서버의 자원을 제어하며 작업을 진행해 나가는 권한이다. 협업서버에 설정된 암호를 바꾸거나 접속환경을 변경하는 작업은 제한된다. 관리자 권한의 협업참여자에 의해 관람자 권한으로 변경 될 수 있다.
- 관람자 권한 : 협업서버의 자원을 제어할 수 없고 작업상황을 단지 모니터링하는 권한이다. 관리자 권한에 의해 작업자 권한으로 변경될 수 있다.

표 1. 접근권한에 따라 허용되는 작업
Table 1. Activities enabled by access authorities

세부 사항 / 권한명	암호 및 서버 설정 변경	접속한 클라이언트 관리	서버 자원 제어	서버 모니터링
관리자 (Admin)	○	○	○	○
작업자 (Worker)	X	X	○	○
관람자 (Spectator)	X	X	X	○

3.1.3. 인증 과정의 확장

UltraVNC 서버에서 사용 권한에 따른 인증 과정을 지원하기 위하여 2.2.절에서 다룬 주요 클래스를 확장하여 협업서버를 개발한다. 각 사용 권한 인증에 필요한 암호를 하나씩 설정할 수 있도록 하며 설정한 암호를 통하여 협업참여자가 인증을 요청하였을 때 협업서버가 이를 처리할 수 있도록 확장한다. UltraVNC 서버의 암호는 윈도우즈 레지스트리에 저장되고 UltraVNC 실행 시 vncserver 클래스의 변수에 복사되어 인증에 사용된다. 따라서 세 가지 암호를 입력받아 레지스트리에 저장할 수 있도록 vncProperties 클래스를 확장한다. 또한 세 가지 암호를UltraVNC 서버 실행 프로세스에서 사용할 수 있도록 vncserver 클래스에 각 권한의 암호를 저장할 수 있는 변수를 마련한다. 다음 [그림 2]는 세 가지 암호를 입력받을 수 있도록 확장된 UltraVNC 서버의 설정 대화상자이며 확장된 vncserver 클래스에서 각 권한의 암호를 저장하는 변수는 [표 2]에서 보이고 있다.

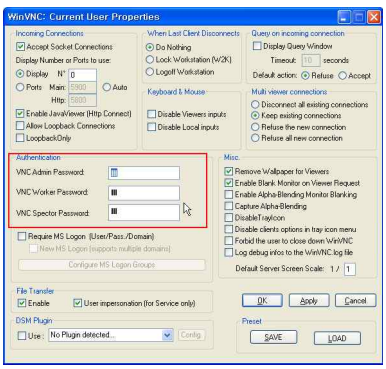

그림 2. 확장된 대화상자
Fig. 2. Extended Dialog

표 2. 각 권한의 암호를 저장하는 vncserver의 변수
Table 2. Variables of vncserver for storing passwords of authorities

이름	역할
m_password	administrator 권한의 암호를 저장
m_fullAccess_password	worker 권한의 암호를 저장
m_viewOnly_password	spectator 권한의 암호를 저장

UltraVNC의 인증은 one-time password 방식[9]을 통하여 클라이언트와 서버간의 전달되는 인증 암호를 암호화하여 보호한다. UltraVNC서버는 접속한 클라이언트에게 임의의 one-time password를 전송하며 세션 객체에도 저장하여 둔다. 인증을 요청하는 클라이언트는 one-time password를 이용하여 사용자가 입력한 암호를 암호화하여 서버에 전송한다. UltraVNC 서버역시 저장해둔 one-time password를 이용하여 서버에 설정된 암호를 암호화 하고 클라이언트가 전송한 암호와 일치여부를 판별하여 인증을 처리한다.

UltraVNC 서버는 환경 설정을 통하여 클라이언트의 입력을 무시하는 View Only기능을 제공하기위하여 vncclient 클래스에 EnablePointer와 EnableKeyboard 메서드를 제공하고 있다. EnablePointer를 통하여 클라이언트가 마우스와 같은 포인터 장치를 이용한 입력의 허용여부를 설정할 수 있으며, EnableKeyboard 메서드를 통하여 키보드를 이용한 입력의 허용여부를 설정할 수 있다.

one-time password를 통한 암호화 방법과 EnablePointer와 EnableKeyboard 메서드를 통하여 정의한 사용권한을 지원할 수 있도록 인증과정을 확장한다. 각 세션객체가 사용권한을 저장할 수 있도록 vncclient 클래스에 m_authmode라는 변수를 마련하였다. 확장된인증과정은 다음과 같은 절차를 따르며 vncClientTread 클래스의 InitAuthenticate 메서드에서 [그림 3]과 같은 코드로 구현되었다. vncClientTread 클래스는 각 세션객체 스레드 처리를 담당하는 클래스이다.

① 클라이언트가 전송한 암호가 관리자 암호와 일치하는지 판별한다.
일치했을 때 : 세션객체의 m_authmode 변수에 관리자 권한 저장, EnablePointer(true), EnableKeyboard(true), 클라이언트에게 rfbVncAuthOK 메시지를 전달.
일치하지 않을 때 : 다음 단계로 진행
② 작업자 암호와 일치하는지 판별한다.
일치했을 때 : 세션객체의 m_authmode 변수에 작업자 권한을 저장, EnablePointer(true), EnableKeyboard(true)
일치하지 않을 때 : 다음 단계로 진행
③ 관람자 암호와 일치하는지 판별한다.
일치했을 때 : 세션객체의 m_authmode 변수에 관람자 권한을 저장, EnablePointer(false), EnableKeyboard(false)
일치하지 않을 때 : 다음 단계로 진행
④ 인증실패를 알리는 VncAuthFailed 메시지를 클라이언트에 전달한다.

```
//클라이언트로부터 암호화된 인증암호를 전송받음
if (!m_socket->ReadExact(response, sizeof(response))) return
FALSE;

//관리자 권한 암호를 one-time password를 통하여 암호화 함
vncEncryptBytes((BYTE *)&challenge, plain);
//사용권한을 관리자 권한으로 설정
m_client->m_authmode = rfbVncAuthOK;
m_client->EnableKeyboard(true);
m_client->EnablePointer(true);

//클라이언트의 암호와 관리자 권한 암호를 비교
for (int i=0; i<sizeof(challenge); i++){
  //관리자 권한 암호와 일치하지 않을 때
  if (challenge[i] != response[i]){
    auth_ok = FALSE;
    //사용 권한을 작업자 권한으로 설정
    m_client->m_authmode = rfbVncAuthWorker;
  }
}
//관리자 권한으로 인증 실패 시 작업자 권한 인증 수행
if(!auth_ok){
  auth_ok = TRUE;
  //작업자 권한 암호를 one-time password을 이용하여 암호화 함
  vncEncryptBytes((BYTE *)&challenge_worker, plain_worker);

  //클라이언트의 암호와 작업자 권한 암호를 비교
  for (int i=0; i<sizeof(challenge_temp); i++){
    //작업자 권한 암호와 일치하지 않을 때
    if (challenge_worker[i] != response[i]){
      auth_ok = FALSE;
      //사용 권한을 관람자 권한으로 설정
      m_client->m_authmode = rfbVncAuthSpector;
      m_client->EnableKeyboard(false);
      m_client->EnablePointer(false);
      break;
    }
  }
}
//관리자 및 작업자 권한으로 인증 실패 시 관람자 권한 인증 수행
if(!auth_ok){
  auth_ok = TRUE;
  //관람자 권한 암호를 one-time password을 이용하여 암호화 함
  vncEncryptBytes((BYTE *)&challenge_spectator,
plain_spectator);
```

```
//클라이언트의 암호와 관람자 권한 암호를 비교
for (int i=0; i<sizeof(challenge_spectator); i++){
  if (challenge_spectator[i] != response[i]){
    auth_ok = FALSE;
    //인증 실패
    m_client->m_authmode = rfbVncAuthFailed;
    break;
  }
 }
}
```

그림 3. 인증과정의 구현
Fig. 3. Implementation of Authentication Process

작업자 권한의 협업참여자는 협업서버의 환경 설정에 관여할 수 없도록 하기위하여 협업서버 환경 설정 대화상자를 띄우거나 서버를 종료하기 전에 관리자 암호를 입력하도록 하였다. 또한 관리자 권한의 협업참여자가 환경 설정 중에는 작업자 권한의 협업참여자가 전송하는 입력을 무시하도록 하였다. 협업서버의 환경 설정이 끝나면 작업자 권한의 협업참여자가 VNC 서버로 전송하는 입력을 허용하도록 한다. 이와 같은 기능을 지원하기 위하여 vncserver 클래스에 작업자 권한의 입력을 모두 거부하도록 하는 SetSpectatorToWorker와 거부된 입력을 허용시키는 ReleaseSpectatorToWorker 메서드를 마련하였다. 두 메서드는 m_authclients 객체에서 작업자 권한의 세션객체를 찾아 EnablePointer와 EnableKeyboard 메서드를 통하여 입력을 활성화 하고 비활성화 하도록 구현하였다.

3.1.4. 사용 권한 관리

관리자 권한의 협업참여자가 권한 관리 작업을 수행할 수 있도록 접근 관리 도구를 구현하였다. 확장된 JavaViewer에는 [그림 4]와 같은 권한관리 도구를 제공한다. 접근 관리 도구는 협업서버에 접속한 모든 협업 참여자의 IP 주소와 권한을 표시하고 작업자 권한과 관람자 권한을 변경하는 기능과 접속을 해제하는 기능을 제공한다. 이러한 기능을 제공하기 위하여 RFB 프로토콜을 확장하여 협업서버가 관리자 권한 클라이언트로 접속자 목록을 전송하는 메시지를 정의하였으며 관리자 권한의 클라이언트가 협업서버로 협업참여자의 권한변경을 요청하는 메시지를 정의하였다. 그리고 확장된 RFB 프로토콜을 지원할 수 있도록 협업서버와 클라이언트 프로그램을 구현하였다.

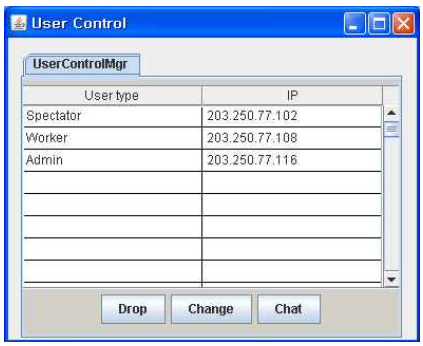

그림 4. 권한 관리 도구
Fig. 4. Access authority administration Tool

3.1.4.1. RFB 프로토콜의 확장

협업서버에서 관리자 권한의 클라이언트에 클라이언트 목록을 전송하는 ClientListMsg 메시지를 RFB 프로토콜의 Server-to-Client 메시지에 정의하였다. ClientListMsg 메시지는 XML 형태로 협업서버에 접속한 협업참여자의 IP 주소와 사용권한 목록을 관리자 권한 클라이언트에 전송한다. 다음 [표 3]은 ClientListMsg 메시지의 구조이며 [그림 5]는 ClientListMsg 메시지가 전송하는 목록의 예이다.

표 3. ClientListMsg 메시지 구조
Table 3. Structure of the ClientListMsg message

No.of bytes	Type	[Value]	Description
1	unsigned char	101	메시지 형식
3			padding
4	unsigned char[4]		XML 길이
length	unsigned char[length]		XML

```
<?xml version="1.0" encoding="euc-kr"?>
<ClientList>
  <Client Type="Admin" IP="123.123.123.2"/>
  <Client Type="Spectator" IP="123.123.123.3"/>
  <Client Type="Worker" IP="123.123.123.4"/>
</ClientList>
```

그림 5. ClientListMsg 메시지의 예
Fig. 5. An Example of ClientListMsg Message

관리자 권한의 클라이언트가 협업서버에 권한변경을 요청하는 rfbAdminRequest 메시지를 Client-to-Server 메시지에 정의하였다. 관리자 권한 클라이언트는 이 메시지를 통하여 변경 대상이 되는 협업참여자의 IP 주소와 변경하고자하는 권한 이름 목록을 XML 형태로 협업서버에 전달한다. 메시지의 구조는

ClientListMsg 메시지와 같으며 협업서버가 메시지를 구별할 수 있도록 message type의 값으로 150을 가진다. 다음 [표 4]와 [표 5]는 확장된 RFB 프로토콜의 Server-to-Client, Client-to-Server 메시지의 종류를 나타내고 있다.

표 4. 확장된 Server-to-Client message의 종류
Table 4. Classification of Extended Server-to-Client messages

No.	Name	비고
0	FramebufferUpdate	갱신된 화면의 프레임 버퍼 전송
1	SetColourMapEntries	색상 맵 정보
2	Bell	비프음을 내도록 함
3	ServerCutText	서버 클립보드의 텍스트 정보
101	ClientListMsg	협업서버의 접속자 목록

표 5. 확장된 Client-to-Server message의 종류
Table 5. Classification of Extended Client-to-Server message messages

No.	Name	비고
0	SetPixelFormat	픽셀의 포맷을 설정
2	SetEncoding	픽셀의 압축 방식을 설정
3	FramebufferUpdateRequest	프레임 버퍼의 갱신 요청
4	KeyEvent	키보드 이벤트
5	PointerEvent	마우스 이벤트
6	ClientCutText	클라이언트 클립보드의 텍스트 정보
150	rfbAdminRequest	사용권한변경 요청

3.1.4.2. 서버 구현

협업서버에서 관리자 권한의 클라이언트에 접속자 목록을 전송하기 위하여 접속한 클라이언트의 IP와 권한 목록을 XML 형태로 작성하도록 하였다. 그리고 접속자 목록을 관리자 권한 클라이언트에 전송해야할 경우를 분석하고 협업서버에서 그러한 경우를 처리하는 클래스에 접속할 수 있는 메서드를 구현하였다. 다음 [표 6]은 관리자 권한 클라이언트에 접속자 목록을 전송하기 위하여 마련한 메서드를 보인다.

표 6. 접속자 목록 전송 메서드
Table 6. Methods for client list transmission

클래스	메서드	기능
vncserver	SendListToAdmin	ListAuthClientsByteArray를 호출하여 접속자목록을 작성하며 관리자 권한 세션객체 sendXMLClientList 메서드를 호출
	ListAuthClientsByteArray	접속자 목록 작성
vncclient	sendXMLClientList	접속자 목록을 ClientListMsg 메시지를 통하여 클라이언트에 전송

관리자 권한 클라이언트에 접속자 목록을 전송해야하는 경우와 협업서버에서 이를 처리하는 부분은 다음 [표 7]과 같다. 각 메서드에서 해당 처리를 마치는 부분에서 vncserver 클래스의 SendListToAdmin 메서드를 호출하여 접속자 목록을 관리자 권한의 클라이언트에 전송할 수 있다.

표 7. 접속자 목록을 전송하는 경우와 처리를 담당하는 부분
Table 7. Client list transmission cases and processing methods

목록을 전송하는 경우	처리 클래스 및 메서드 (클래스::메서드)
관리자 권한의 클라이언트가 협업서버에 인증 성공한 경우	vncclient::InitAuthenticate
협업서버에 접속한 클라이언트가 인증을 성공한 경우	vncclient::InitAuthenticate
협업서버의 접속자가 접속을 끊은 경우	vncclient::Kill
권한을 변경한 경우	vncserver::ChangeUserAuth

vncserver 클래스의 ChangeUserAuth 메서드는 관리자 권한의 클라이언트가 전송한 rfbAdminRequest 메시지를 분석하고 권한변경을 수행하기 위하여 구현한 메서드이다. rfbAdminRequest 메시지는 XML 형태로 권한변경 정보를 협업서버에 전달한다. 다음 [그림 6]에서 보이는 rfbAdminRequest 메시지는 IP 주소가 123.123.123.1인 협업참여자의 권한을 작업자 권한에서 관람자 권한으로 변경하며, 123.123.123.3인 협업참여자의 권한을 관람자 권한

에서 작업자 권한으로 변경하고 123.123.123.6인 협업참여자의 접속을 끊도록 요청하는 예를 보이고 있다. ChangeUserAuth 메서드는 rfbAdminRequest 메시지를 MSXML Parser[10]를 이용하여 분석한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClientList>
        <Client Type="Spectator" IP="123.123.123.1"/>
        <Client Type="Worker" IP="123.123.123.3"/>
        <Client Type="Drop" IP="123.123.123.6"/>
</ClientList>
```

그림 6. rfbAdminRequest 메시지의 XML 내용의 예
Fig. 6. An example of XML in a rfbAdminRequest message

분석을 통하여 얻어진 IP를 이용해 세션객체의 m_authmode 변수에 요청된 권한을 저장한다. 그리고 세션객체의 EnablePointer와 EnableKeyboard 메서드를 통하여 입력허용 또는 거부를 설정한다.

UltraVNC 서버의 vncclient 클래스는 클라이언트의 접속해제를 처리하는 메서드로 Remove와 Kill 메서드를 제공하고 있다. Remove 메서드는 세션객체 스스로 소멸을 요청할 때 쓰이며, Kill 메서드는 다른 객체가 세션객체의 소멸을 요청할 때 쓰인다. 따라서 vncserver 객체의 ChangeUserAuth 메서드는 Kill 메서드를 통하며 세션객체의 소멸을 요청하여 해당 클라이언트의 접속을 끊는다.

3.1.4.3. 클라이언트 구현

클라이언트가 관리자 권한 인증을 성공하면 협업서버로부터 VncAuthOk 메시지를 전송받고 ClientListMsg 메시지를 전송받는다. JavaViewer의 RfbProto 클래스와 ButtonPanel 클래스를 확장하여 VncAuthOk 메시지를 전송받았을 때 접근관리 도구 실행 버튼을 활성화하도록 하였다.

접근 관리 도구는 ClientListMsg 메시지에 전송된 접속자 목록을 표시할 수 있는 기능과 rfbAdminRequest 메시지를 통하여 권한 변경 요청을 협업서버에 전송하는 기능을 제공하며 이를 위하여 다음 [표 8]에서 보이는 클래스가 구현되었다.

표 8. 접근 관리 도구의 클래스
Table 8. Classes for access authority administration tool

클래스	역할
UserCont-rolFrame	접근 관리 도구 대화상자 클래스이며 접속자 목록을 javax.swing.JTable[11]을 이용하여 표시한다. 접속자를 선택하여 권한 변경 및 접속 해제 할 수 있는 인터페이스를 제공하며 rfbAdminRequest 메시지를 통하여 전송할 XML형태의 권한 변경 요청 정보를 작성한다.
ClientList-XmlParser	UserControlFrame에 표시할 ClientListMsg메시지의 접속자 목록을 분석한다. 접속자 목록은 XML 형태이므로 SAX 파싱을 지원하는 org.xml.sax.ContentHandler[12]을 이용하였다.

UserControlFrame은 [그림 4]와 같은 대화상자를 표시하며 목록에서 선택한 사용자의 접속 해제를 요청 하는 'Drop' 기능과 권한 변경을 요청하는 'Change' 기능을 제공한다. 관리자 권한에 대한 접속 해제와 권한 변경 요청은 할 수 없으며, 작업자 권한과 관람자 권한을 상호 변경하는 요청과 접속해제 요청을 지원한다. UserControlFrame에서 작성한 권한 변경 요청을 rfbAdminRequest 메시지를 통하여 협업서버에 전송할 수 있도록 JavaViewer의 RfbProto 클래스를 확장하였다.

3.1.4.4. 시나리오별 실행 흐름

다음 [그림 7]은 협업서버와 확장된 JavaViewer를 통한 실시간 협업에서 협업을 방해하려는 목적의 접속자의 접속을 해제하는 모습을 보이고 있다. 123.123.123.2라는 IP 주소를 가지는 관리자 권한의 협업참여자가 접근 관리 도구를 통하여 악의적인 목적의 접속자의 IP 주소와 Drop 요청을 rfbAdminRequest 메시지를 통하여 협업서버에 전송한다. 협업서버의 vncserver 클래스는 IP 주소를 참조하여 해당 세션객체를 찾고 세션객체의 Kill 메서드를 호출하여 접속을 해제한다.

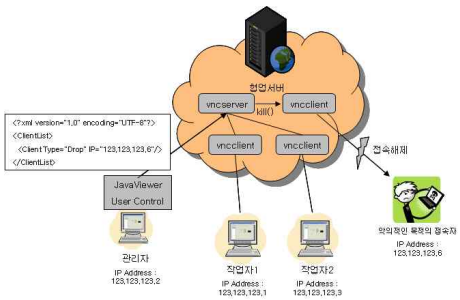

그림 7. 접속 해제 시나리오
Fig. 7. Disconnection Scenario

다음 [그림 8]은 관리자 권한의 협업참여자에 의하여 작업자1과 작업자2의 권한 변경 사례를 보이고 있다. 작업자 1은 작업자 권한이며 작업자 2는 관람자 권한이다. 관리자 권한의 협업참여자는 접근 관리 도구를 이용하여 작업자 1을 관람자 권한으로 작업자 2를 작업자 권한으로 변경하는 요청을 하는 rfbAdminRequest 메시지를 협업서버에 전송한다. 협업서버의 vncserver 클래스는 rfbAdminRequest 메시지의 IP 주소를 참조하여 작업자 1과 작업자 2의 세션객체를 찾고 각 세션객체의 EnablePointer와 EnableKeyboard 메서드를 통하여 입력 허용을 설정한다.

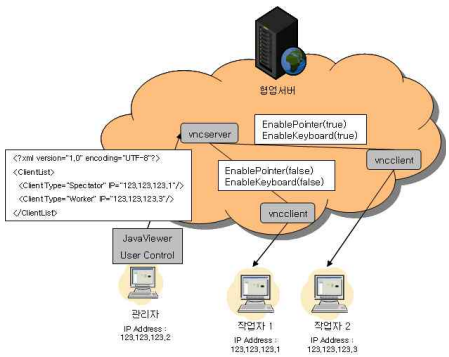

그림 8. 권한 변경 시나리오
Fig. 8. Access authority modification scenario

3.2. 원클릭 접속을 위한 환경

JavaViewer 클라이언트를 사용하여 VNC 서버에 접속하기 위해서는 서버의 IP 주소와 VNC 서버가 사용하는 포트를 알아야 한다. UltraVNC 서버는 원격시스템의 화면을 전송하기 위하여 다양한 옵션과 인증 방식을 제공하고 있지만, 협업 참가자들이 협업서버에 접속하기 위해 그러한 설정을 매번 확인하는 것은 매우 번거로운 과정이며, VNC에 익숙하지 못한 협업서버 관리자나 협업 참가자들에게 어려움을 준다.

UltraVNC의 원격 제어 기능을 유지하면서 손쉬운 접속 방법을 제공하여 위와 같은 문제점을 해결할 수 있도록 원클릭 접속 환경을 구현하였다. 원클릭 접속환경을 이용하면 VNC를 CoSpace[13]와 같은 다른 협업시스템과 손쉽게 연동하는 것도 가능하다. 다음 [그림 9]는 원클릭 접속 환경을 위한 다음과 같은 시나리오를 보이고 있다.

협업서버는 서버의 환경을 저장하는 파일을 생성한다.
서버 환경 설정 파일을 웹이나 ftp와 같은 서비스를 통하여 게시한다.
협업참여자는 게시된 환경 설정 파일을 다운로드 한다.
환경 설정 파일의 확장자에 대한 연결프로그램을 로컬 컴퓨터의 운영체제에서 지정할 수 있다. 환경 설정 파일의 확장자의 연결프로그램으로 확장된 JavaViewer를 지정한다.
서버 환경 설정 파일을 클릭하는 것으로 확장된 JavaViewer를 실행할 수 있으며 협업서버에 접속할 수 있다.

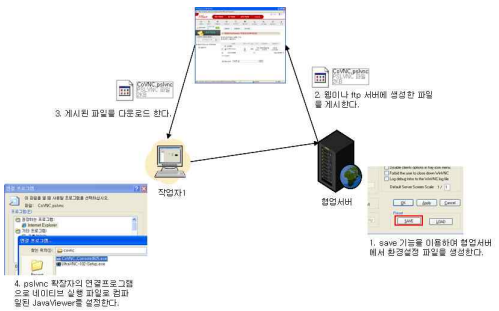

그림 9. 원클릭 접속 시나리오
Fig. 9. One click connection scenario

3.2.1. 협업서버의 환경 설정 저장 파일 구현

UltraVNC 서버는 [그림 2]와 같은 환경 설정 대화상자를 제공한다. 환경 설정 대화상자는 환경 설정에 필요한 각각의 기능을 활성화하거나 비활성화할 수 있는 트리 구조의 컴포넌트로 구성되어있다. 따라서 서버 설정 대화 상자가 각 컴포넌트의 ID와 설정 상태를 확인하여 서버 환경 설정을 저장하는 파일을 생성하도록 구현하였다. 설정 저장 파일은 설정 정보를 XML 형태로 저장한다.

UltraVNC 서버의 환경 설정 대화 상자는 vncProperties 클래스에 구현되어 있다. vncproperties 클래스에서 대화상자에서 일어나는 이벤트는 DialogProc 메서드[14]에서 처리한다. 환경 설정 대화 상자에 다음 [그림 10]과 같이 설정을 파일에 저장할 수 있는 'SAVE' 버튼과 설정을 파일로부터 불러올 수 있는 'LOAD' 버튼을 추가하였다.

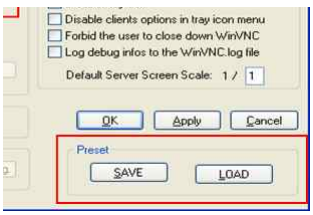

그림 10. 서버 설정 대화상자에 추가된 버튼
Fig. 10. Added buttons at a server properties dialog

추가한 버튼의 이벤트를 처리하기 위하여 DialogProc 메서드를 확장하였다. 저장 이벤트가 발생하였을 때, 대화상자의 각 컴포넌트의 ID와 설정 값을 바탕으로 MSXML Parser를 이용하여 XML 형태의 저장 파일을 작성하도록 확장하였다. 그리고 불러오기 이벤트가 발생 하였을 때, 저장 파일을 MSXML Parser를 통하여 분석하여 얻은 컴포넌트의 ID와 설정 값을 대화상자에 적용하도록 하였다. 단 각 권한을 인증하기 위한 암호 컴포넌트 ID와 설정된 암호는 저장하지 않는다.

서버 설정 파일을 통하여 클라이언트가 협업서버에 접속할 수 있도록 설정 파일에 협업서버의 IP주소를 저장한다. UltraVNC 서버는 클라이언트와 통신을 위하여 WinSock[14]을 이용한 소켓 통신을 한다. 이러한 소켓 통신은 vsocket 클래스가 담당하고 있다. 이 클래스에 소켓이 사용하는 서버의 IP 주소를 얻어 서버 설정 파일에 저장할 수 있다. 따라서 vsocket 클래스에 IP 주소를 반환하는 기능의 GetAddress 메서드를 구현하였다. 그리고 vncPropertise 클래스의 DialogProc 메서드에서 저장 이벤트를 처리할 때 구현한 GetAddress 메서드를 호출하여 서버의 IP 주소를 서버 설정 파일에 저장하도록 하였다.

서버 환경 설정파일의 확장자는 윈도우즈 플랫폼 환경에서 이 파일을 더블 클릭 했을 때의 사용할 연결 프로그램을 지정할 수 있도록 'xml'이 아닌 'vnc'으로 하였다. 다음 [그림 11]은 생성된 서버 환경 설정파일내용의 예이다.

```
<?xml version="1.0" encoding="euc-kr" ?>
<UltraVNC_preset>
  <server_ip>203.250.77.117</server_ip>
<IDC_CONNECT_BORDER>
  <IDC_CONNECT_SOCK>1</IDC_CONNECT_SOCK>
  <IDC_SPECDISPLAY>0</IDC_SPECDISPLAY>
  <IDC_DISPLAYNO />
  <IDC_SPECPORT>1</IDC_SPECPORT>
  <IDC_PORTRFB>5901</IDC_PORTRFB>
  <IDC_PORTHTTP>5800</IDC_PORTHTTP>
  <IDC_PORTNO_AUTO>0</IDC_PORTNO_AUTO>
  <IDC_CONNECT_HTTP>1</IDC_CONNECT_HTTP>
  <IDC_ALLOWLOOPBACK>0</IDC_ALLOWLOOPBACK>
  <IDC_LOOPBACKONLY>0</IDC_LOOPBACKONLY>
</IDC_CONNECT_BORDER>
<Multi_viewer_connections>IDC_MV1</Multi_viewer_connect
ions>
...
</UltraVNC_preset>
```

그림 11. 서버 환경 설정파일의 예
Fig. 11. An Example of the Server Configuration File

3.2.2. JavaViewer 확장을 통한 원클릭 접속 지원

확장된 JavaViewer는 세 가지 방법을 통하여 협업서버에 접속할 수 있다. 첫째로 기존 JavaViewer의 접속방법과 같이 실행인자로 서버의 IP 주소와 포트를 입력하는 것이다. 둘째로 서버 환경 설정파일의 경로를 실행인자로 입력하여 접속하는 것이다. 셋째로 실행인자를 입력하지 않고 javax.swing.JFileChooser 클래스[15]를 통하여 서버 환경 설정파일 선택하여 접속하는 방법이다. 이와 같은 접속 방법을 지원하기 위하여 JavaViewer의 실행 클래스인 VncViewer 클래스를 다음 [그림 12]와 같은 접속 절차를 가지도록 확장하였다.

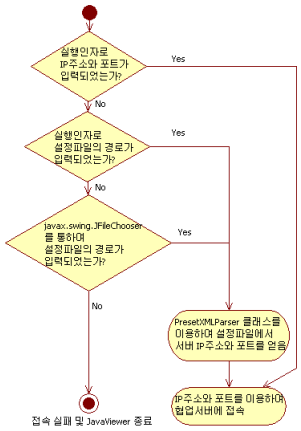

그림 12. 확장된 JavaViewer의 접속절차
Fig. 12. Connection process of the extended JavaViewer

PresetXMLParser 클래스는 org.xml.sax.ContentHandler 인터페이스를 통하여 설정 파일을 분석하는 기능을 한다. VncViewer 객체는 PresetXMLParser 객체를 통하여 설정 파일에 저장된 서버 IP 주소와 포트를 얻고 이를 통하여 협업서버에 접속한다.

IV. 평가

Real VNC와 Tight VNC, Ultra VNC와 같은 다른 오픈소스 VNC와 확장 VNC의 협업 지원기능에 대하여 비교한다. Real VNC는 AT&T에서 최초로 개발된 VNC를 지칭하며 VNC를 개발한 연구원들에 의해 현재에도 구입하여 사용할 수 있는 버전과 오픈소스 무료 버전을 제공하고 있다. 비교를 위하여 사용된 Real VNC의 오픈소스 버전은 사용할 수 있는 서버의 플랫폼으로 윈도우즈, 리눅스를 지원하고 있으며 클라이언트의 플랫폼은 제약이 없다[6]. Tight VNC는 오픈소스 VNC 소프트웨어로 Real VNC와 비슷한 기능을 제공

하고 있으며, JPEG와 같은 정지화면 인코딩 기술을 통하여 압축한 서버의 화면을 클라이언트에 전송하는 기능을 제공한다. 비교를 위하여 고려한 기능은 다음과 같다.

다중세션 지원 : 협업을 위하여 협업 시스템은 여러 접속자가 같은 권한 혹은 다른 권한으로 협업 서버에 동시에 접속할 수 있는 다중세션 환경을 지원하여야 한다.

다중 권한 지원 : 효율적인 작업진행을 위하여 실제로 저작작업을 수행하는 접속자와 수행하지 않는 접속자에 대한 권한을 지원하여야 한다.

권한 관리 지원 : 다중 권한을 관리자 권한 접속자가 클라이언트 소프트웨어를 이용하여 관리할 수 있어야 한다.

다중 플랫폼 지원 : 언제 어디서든 클라이언트가 협업 서버에 접속할 수 있도록 클라이언트 소프트웨어가 다양한 플랫폼을 지원해야 한다.

간편한 접속 절차 지원 : 협업서버 접속에 필요한 노력을 최대한 줄이고 다양한 공동 작업 환경에 적용할 수 있는 기능을 지원하여야 한다.

확장 VNC는 [표 9]에서 보이는 것과 같이 효율적인 협업을 지원하기 위한 기능을 모두 충족하는 것으로 평가할 수 있다.

표 9. 협업 지원 기능 비교
Table 9. Comparison of supported collaboration facilities

원격제어 도구 / 기능	확장 VNC	Real VNC	Tight VNC	Ultra VNC
다중세션 지원	O	O	O	O
다중 권한 지원	O	X	X	X
권한 관리 지원	O	X	X	X
다중 플랫폼 지원	O	O	O	O
간편한 접속 절차 지원	O	X	X	X

협업 지원을 위하여 추가한 ClientListMsg와 rfbAdminRequest 메시지는 각각 특별한 이벤트가 발생할 때 사용되어진다. ClientListMsg 메시지는 항상 전송되는 것이 아니라 특정 협업참여자가 협업 서버에 접속하거나 접속을 해제하였을 때, 또는 관리자 권한의 협업참여자가 특정 참여자의 권한을 변경했을 때에만 전송된다. 또한 rfbAdminRequest 메시지 역시 관리자 권한의 협업 참여자가 협업 서버에 권한 변경을 요청할 때만 사용된다. 각 메시지가 전송하게 되는 XML형식의 목록은 그 크기가 정해진 것은 아니나, 접속자 하나를 표현하기 위하여 최대 50bytes 정도만을 사용하므로, 확장 전의 VNC가 필요로 하는 메시지의 양과 거의 차이가 없다고 말할 수 있다.

V. 결론

그림 또는 음악 등의 콘텐츠를 저작하는 작업은 GUI 환경을 통하여 진행되며 고가의 응용 소프트웨어가 필요하다. Thin Client 컴퓨팅 소프트웨어인 VNC는 GUI 환경을 통하여 서버 기반 컴퓨팅을 지원하고 있다. 따라서 VNC를 이용한 협업 작업은 저작 환경의 접근성, 비용, 보안 및 관리, 유지관리 등의 장점을 가진다. 현재 배포되고 있는 CoSpace와 같은 협업 지원 시스템은 서버/클라이언트 기반의 비동기식 협업 작업을 지원하고 클라이언트에 저작 작업에 필요한 환경이 모두 갖추어져야 하는 Fat Client 시스템이다. 따라서 Thin Client 컴퓨팅의 장점을 협업 시스템에 반영하고, GUI를 통한 실시간 협업작업을 지원하기 위하여 VNC를 확장하였다.

확장된 UltraVNC 서버와 JavaViewer 클라이언트 프로그램은 원격 제어를 통해 자원을 제공하는 시스템을 협업서버로 구성하고, 협업참여자들이 협업서버를 원격 제어함으로써 협업 작업을 진행할 수 있는 기능을 제공한다. 또한 확장 VNC는 서버 환경 설정파일을 통해 VNC 서버를 이용한 다양한 협업서버 환경을 손쉽게 구성하는 기능을 제공하고 있으며, 서버 환경 설정파일을 한 번만 클릭하여 손쉬운 협업서버 접속 기능을 제공하고 있다. 그리고 이러한 기능을 바탕으로 CoSpace와 같은 협업기능을 지원하는 프로그램에서 협업 VNC 서버로 마련된 동기식 협업서버를 간편하게 이용할 수 있는 기능을 구현할 수 있다. 또한 원격 제어를 목적으로 만들어진 RFB 프로토콜에 협업작업자 관리 기능을 확장하고, 협업서버와 JavaViewer 클라이언트는 확장된 RFB 프로토콜을 바탕으로 한 협업참여자 관리기능을 제공하여 실시간으로 진행되는 협업 작업을 신뢰성 있게 수행할 수 있도록 지원

할 수 있다.

차후 비동기식 협업과 동기식 협업을 모두 지원할 수 있는 컴퓨터 공동 작업 환경을 제공하기 위하여, WebDAV기반의 CoSpace 협업 시스템과 확장 VNC를 연동하는 방법을 연구할 예정이며, RFB 프로토콜의 비효율적인 요소를 찾고 이를 개선하여 좀 더 반응성이 나은 원격제어 프로토콜에 대하여 연구할 예정이다.

참고문헌

[1] Thin Client Computing, "What is Thin Client Computing?", http://www.thinclient.net/whatis_thinclient.html

[2] 류한석, "Server Based Computing", 한국소프트웨어진흥원 SW Insight 정책리포트, 2007.5

[3] QNB Intelligence, "Server-Based and Thin Client Computing", 2003

[4] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, "Virtual Network Computing", 1998

[5] Tristan Richardson, "The RFB Protocol", 2006

[6] RealVNC, "RealVNC", http://www.realvnc.com

[7] TightVNC, "TightVNC", http://www.tightvnc.org

[8] UltraVNC, "UltraVNC", http://www.uvnc.com

[9] Behrouz A Forouzan, "Cryptography & Network Security", 2007

[10] Microsoft MSDN, "XML 문서 및 데이터", http://msdn.microsoft.com/ko-kr/library/2bcctyt8.aspx

[11] Sun Java2SE API, "javax.swing.JTable" http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JTable.html

[12] Sun Java2SE API, "org.xml.sax.ContentHandler" http://java.sun.com/j2se/1.4.2/docs/api/org/xml/sax/ContentHandler.html

[13] 신원준, 박희종, 김동호, 박양수, 이명준, "WebDAV기반 협업시스템의 클라이언트 개발", 한국정보과학회 춘계학술대회, Vol. 2005

[14] Charles Petzold, "Programming Windows, 5th Edition", 2006

[15] Sun Java2SE API, "javax.swing.JFileChooser" http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JFileChooser.html

저 자 소 개

이 태 호

2008년 2월 : 울산대학교 컴퓨터정보통신공학부 졸업

2008년 3월 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 석사과정

관심분야 : 협업시스템, 원격 컴퓨팅, 홈네트워크, 분산프로그래밍

이 홍 창

2008년 2월 : 울산대학교 컴퓨터정보통신공학부 석사

2008년 3월 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 박사과정

관심분야 : 웹기반 정보시스템, 협업시스템

박 양 수

1978년 : 울산대학교 전산학과 졸업 (공학사)

1981년 : 서울대학교 계산통계학과 졸업 (이학석사)

1986년 : 서울대학교 계산통계학과 박사과정 수료

1980년 ~ 현재 : 울산대학교 정보통신공학부 교수

관심분야 : 분산처리, 컴퓨터알고리즘

이 명 준

1980년 : 서울대학교 수학과 졸업(학사)

1982년 : 한국과학기술원 전산학과 졸업(석사)

1991년 : 한국과학기술원 전산학과 졸업(박사)

1982년 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 (교수)

1993년 ~ 1994년 : 미국 버지니아 대학 교환교수

2005년 ~ 2006년 : 미국 캘리포니아 주립대학 교환교수

관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산프로그래밍, 생물정보학, 센서 네트워크 프로그래밍 환경