
CoDisplay : 대형 디스플레이 장치를 활용하기 위한 VNC 기반 회의도구

이태호* · 박지혜* · 이기훈** · 이명준***

CoDisplay : A Conference Tool Based on VNC for Utilizing Wide Display Devices

Taeho Lee* · Ji-Hye Park* · Ki-Hoon Lee** · Myung-Joon Lee***

이 논문은 2008년 울산대학교 연구비에 의하여 연구되었음.

요 약

최근 대형 디스플레이 장치를 활용한 의사전달행위가 보편화되고 있다. 따라서 효과적인 의사전달을 지원할 수 있도록 대형 디스플레이 장치를 효율적으로 공유하는 것이 바람직하다. VNC(Virtual Network Computing)는 데스크톱 공유 도구로서 다른 지역에 위치한 컴퓨터로 데스크톱 화면을 전송할 수 있다. 또한 VNC를 이용하여 두 명 이상의 사용자가 동시에 같은 데스크톱 화면을 사용할 수 있다. 그러나 현재 VNC가 지원하는 기능만으로 대형 디스플레이 장치를 효율적으로 공유하는 것은 어려운 일이다.

본 논문에서는 VNC를 이용하여 대형 디스플레이 장치를 공유하는 회의도구인 CoDisplay 개발에 대하여 기술한다. 회의 참가자들은 네트워크 주소를 이용하여 대형 디스플레이 장치에 접속하여 자신의 데스크톱 화면을 내장된 VNC를 이용하여 전송한다. 또한 여러 데스크톱 화면을 배열하고 전환하는 기능을 통하여 대형 디스플레이 장치를 효과적으로 공유할 수 있다.

ABSTRACT

Since wide display devices have recently become popular for communication activity, it is desirable for effective communication to share those displays in an efficient way. VNC(Virtual Network Computing) is a desktop sharing software, which transmits the desktop screen and the related events to other computers on different places. So, through VNC, it is possible for more than two users to share the same desktop screen at the same time. But it is very cumbersome to share wide display devices through the functions currently supported by VNC.

In this paper, we describe the development of CoDisplay which is a VNC-based conference tool for efficient sharing of a wide display device. Conference participants connect to the wide display device through the associated network address, and transmit their desktop screen to the device. The control functions for the participants support the division of the display screen for multiple desktop screens and the switchover of desktop screens.

키워드

대형 디스플레이 장치(Wide display devices), VNC, 회의도구(A conference tool), CoDisplay

* 울산대학교 컴퓨터정보통신공학과 석사과정

접수일자 2009. 02. 10

** 울산대학교 컴퓨터정보통신공학부 학사과정

심사완료일자 2009. 02. 27

*** 울산대학교 컴퓨터정보통신공학부 교수 (교신저자)

I. 서 론

최근 디스플레이 장치 제조기술의 발전에 따라 넓은 화면을 제공할 수 있는 대형디스플레이 장치가 여러 분야에서 널리 보급되어 사용되고 있다. 특히 여러 사람이 의견을 주고받는 회의나 강연과 같은 분야에서 대형 디스플레이 장치와 이에 영상을 전달하는 셋톱박스를 많이 이용하고 있다. 이것은 효과적인 의사전달을 위하여 구두 발표 이외에도 동영상 자료나 프레젠테이션 문서 등과 같은 다양한 콘텐츠를 활용하기 때문이다. 또한 오늘날에는 과거와 달리 콘텐츠를 쉽게 저작할 수 있는 환경이 마련되어 있다. 예를 들어 회의 참여자는 시각적 자료를 저작하기 위하여 웃사전n O저작할 e[1]나 Google 오피스[2]와 같은 기능이 우수한 도구를 쉽게 구하여 사용할 수 있다. 따라서 회의 참여자 또는 강연자 및 청강자들은 마련된 대형 디스플레이 장치를 이용하여 자신이 준비한 콘텐츠 저작물을 이용하여 의사를 효과적으로 표현할 수 있다.

대형 디스플레이 장치에 저작물을 출력하는 가장 기본적인 방법으로 노트북 PC와 같은 기기를 대형 디스플레이 장치에 유선으로 연결하는 방법이다. 유선 연결은 흔히 D-Sub나 Composite 등과 같은 유선 비디오 연결을 이용한다. 그러나 단지 하나의 화면만을 보여주므로 빈번한 의사교환이 발생하여 여러 사람이 디스플레이 장치를 번갈아가며 사용할 때 연결설치 및 철거 작업이 번거로워지는 문제가 있다. 두 번째로 대형 디스플레이 장치와 PC를 연결하고 여러 발표자들의 발표하고자하는 저작물을 USB 메모리 등의 소형저장장치에 담아 연결된 PC에서 실행하는 방법이 있다. 소형 저장 장치는 연결 및 제거가 간편하여 유선 연결 방법이 가지는 문제점을 해결할 수 있다. 그러나 저작물을 저작하고 테스트하는 환경과 실제 출력하는 환경이 다를 수 있으므로 정상적인 출력을 보장할 수는 없다. 또한 다양한 콘텐츠를 모두 출력할 수 있는 환경을 구축하는 비용은 큰 부담이 된다. 따라서 두 가지 방법의 문제점을 해결할 수단이 필요하다.

VNC [3]는 TCP/IP 네트워크를 통하여 데스크톱 화면을 공유하는 기능을 제공한다. 따라서 대형 디스플레이 장치와 회의참여자의 노트북 PC가 네트워크로 연결되는 환경에서 VNC의 데스크톱 화면 공유 기능을 사용할 수 있다. VNC는 화면을 송출하는 서버와 그

것을 수신하는 클라이언트로 구성된다. 따라서 화면을 전송하고자하는 회의 참여자는 자신의 VNC 서버에 대형 디스플레이 장치와 연결된 PC의 VNC 클라이언트가 접속할 수 있도록 하여야한다. 그러나 VNC 서버는 VNC 클라이언트가 자신에게 접속하도록 요청하는 기능을 제공하지 않는다. 또한 VNC 클라이언트는 여러 창을 효율적으로 제어하는 기능을 제공하지 않아 빈번한 의사교환이 발생했을 때 문제점을 해결할 수 없다.

본 논문에서는 VNC를 이용하여 빈번한 의사전달이 오고가는 회의 분야에서 대형 디스플레이 장치를 회의 참여자들이 효과적으로 공유할 수 있도록 하는 CoDisplay에 대하여 기술한다. CoDisplay를 통하여 사용자는 간단히 네트워크상의 주소를 입력하여 대형디스플레이 장치에 데스크톱 화면을 출력 할 수 있다. 또한 클라이언트 기반의 간편한 인터페이스를 통하여 대형 디스플레이 장치에 표시할 사용자의 화면을 간편하게 전환할 수 있는 기능을 제공하여 설치 및 철거 작업에 따른 문제점을 해결한다. 또한 화면 분할 기능을 제공하여 여러 의견들을 하나의 화면에서 비교 분석할 수 있도록 도와준다.

본 논문의 구성은 다음과 같다. 2절에서 화면을 전송하기 위하여 사용된 VNC를 소개한다. 3절에서는 효과적인 회의 진행을 돋기 위한 CoDisplay의 인터페이스에 대해 기술하며 기술한 인터페이스 및 기능을 제공하기 위한 설계 및 개발에 대하여 기술한다. 4절에서는 개발한 CoDisplay의 효율성에 대한 결론 및 향후 연구 과제를 제시한다.

II. 관련 연구

2.1. VNC

CoDisplay는 대형 디스플레이 장치에 화면을 전송하기 위하여 내부적으로 VNC 소프트웨어를 사용한다. VNC는 특정 시스템의 데스크톱 화면을 공유하여 원격 제어를 수행할 수 있는 기능을 제공하는 Thin Client 컴퓨팅[4] 소프트웨어이다. VNC는 데스크톱 화면을 전송하는 서버와 수신하는 클라이언트로 구성된다. 서버와 클라이언트는 통신을 위하여 TCP/IP 기반의 RFB 프로토콜[5]을 이용한다.

현재 배포되고 있는 VNC는 다양한 플랫폼과 특별한 기능을 지원하기 위하여 다양한 형태의 서버와 클라이언트 형태를 지니고 있다. 대표적인 VNC 소프트웨어로는 Real VNC[6], Tight VNC[7], Ultra VNC[8] 등이 있다. 또한 VNC 소프트웨어는 오픈소스 형태로 제공되므로 VNC의 원격 제어 기능을 기반으로 특별한 작업을 수행할 수 있도록 VNC와 RFB 프로토콜을 확장하여 사용할 수 있다. 그 예로 협업을 지원할 수 있는 협업 VNC[9]와 SSL 암호화 연결을 지원하는 Secure VNC[10] 등이 있다. 그러나 대형 디스플레이 장치를 공유하여 효과적인 의사전달을 돋도록 하는 VNC는 존재하지 않는다.

2.2. UltraVNC 서버와 Mirror Video Driver

Ultra VNC 서버는 오픈소스 VNC 서버의 한 종류로서 윈도우즈 기반의 시스템을 GUI를 통하여 효과적으로 원격 제어할 수 있는 기능을 제공한다. 대표적인 기능으로 Mirror Video Driver[8,9,11], 윈도우즈 계정 로그인 기능, 메시지 암호화, 서버와 클라이언트간의 채팅 및 파일 전송 기능을 제공하고 있다.

Ultra VNC 서버가 제공하는 Mirror Video Driver 기능은 비디오 드라이버의 프레임버퍼 메모리를 Ultra VNC 서버에서 사용할 수 있도록 하는 역할을 한다. Mirror Video Driver는 비디오 드라이버의 프레임버퍼 메모리에 변경된 사항을 즉시 가져와 Ultra VNC 서버에 전달하므로 빠른 반응속도와 적은 CPU 사용량을 보장한다. 따라서 회의도구의 효율적인 화면전송을 위하여 Ultra VNC 서버를 사용하며 Mirror Video Driver 기능을 이용한다.

2.3. JavaViewer

JavaViewer[8,9]는 자바 언어로 개발된 오픈소스 VNC 클라이언트이다. 따라서 자바 가상 머신을 동작할 수 있는 모든 환경에서 실행이 가능한 장점이 있다. 또한 JavaViewer는 C++ 언어로 개발된 플랫폼 전용 클라이언트와 동일한 기능을 제공한다. 다음 [표 1]은 JavaViewer의 주요 클래스와 역할을 보이고 있다.

표 1. JavaViewer의 주요 클래스와 역할
Table 1. Major classes of JavaViewer and their actions

클래스	역할
VncViewer	JavaViewer를 실행하기 위한 main 메서드와 init 메서드가 있으며 JavaViewer의 각 캐쳐와 스레드를 생성한다. JavaViewer 원도우 이벤트를 처리한다.
RfbProto	소켓을 통하여 Vnc 서버와 RFB프로토콜 메시지를 주고받는다.
VncCanvas	전송받은 데스크톱 화면을 표시하며 사용자의 입력을 감지한다.
ClipboardFrame	원격시스템의 클립보드에 전송할 텍스트를 입력하거나 원격시스템으로부터 전달 받은 클립보드 텍스트를 표시한다.

III. CoDisplay의 개발

CoDisplay는 서버와 클라이언트로 구성된다. 서버는 클라이언트로부터 송출된 화면을 출력하는 출력부와 출력부를 클라이언트의 요청에 따라 제어하는 제어부로 구성된다. 클라이언트는 데스크톱 화면을 송출하는 송출부와 출력 화면을 제어하는 요청을 CoDisplay 서버로 전송하기 위한 인터페이스로 이루어진다. 다음 [그림 1]은 CoDisplay를 이용하여 회의 참여자들이 대형디스플레이 장치를 공유하는 예이다.

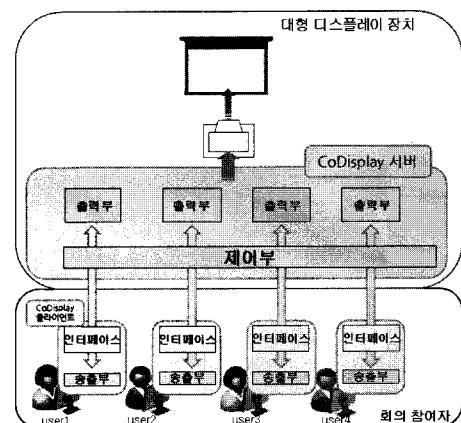


그림 1. CoDisplay를 이용한 대형 디스플레이 장치 공유의 예

Fig. 1 An example of sharing an wide display device using CoDisplay

3.1. CoDisplay 클라이언트 기반 인터페이스의 설계

CoDisplay는 클라이언트에서 서버에 접속하여 화면을 송출하도록 하고 여러 출력 화면을 간편하게 제어할 수 있도록 하는 인터페이스를 가진다. 다음은 클라이언트가 제공하는 각 인터페이스이다.

- 회의 참여자가 단순히 대형 디스플레이 장치의 네트워크 주소나 이름을 입력하여 대형 디스플레이 장치의 CoDisplay 서버에 연결하고 화면을 출력할 수 있도록 [그림 2]에서 보이는 인터페이스
- 대형 디스플레이 공유자 목록에서 공유자를 선택하여 선택된 공유자의 화면을 활성화하는 [그림 3]과 같은 인터페이스
- 대형 디스플레이 공유자 목록에서 특정 공유자의 화면을 지정한 크기로 조절할 수 있도록 하는 인터페이스
- 여러 공유자의 화면을 바둑판식으로 자동 배열하는 기능과 이를 한번의 클릭으로 수행할 수 있는 인터페이스

위와 같은 인터페이스는 [그림 2]와 같이 윈도우즈의 트레이 아이콘을 통하여 사용할 수 있도록 한다.

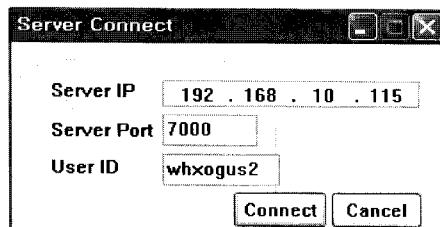


그림 2. CoDisplay 서버에 접속하는 CoDisplay 클라이언트 인터페이스

Fig. 2 The interface of CoDisplay client for connection to the CoDisplay server

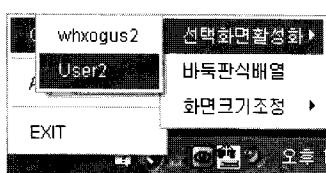


그림 3. 선택된 화면을 활성화하는 CoDisplay 클라이언트 인터페이스

Fig. 3 The interface of CoDisplay client to activate the selected screen

3.2. CoDisplay의 구조

CoDisplay의 서버는 클라이언트가 설계한 인터페이스에 따라 요청한 명령을 처리한다. 또한 서버는 처리된 결과를 클라이언트에 전송한다. 따라서 서버와 클라이언트가 사용하는 프로토콜을 설계한다. 다음 [표 2]는 서버에서 클라이언트로 처리결과를 전송하는데 사용되는 프로토콜 메시지며, [표 3]은 클라이언트에서 서버로 요청하는데 사용되는 프로토콜 메시지이다.

표 2. 서버에서 클라이언트로 전달되는 메시지
Table 2. Transmitted Messages from a Server to a Client

메시지	역할
S2C_ACCEPT	클라이언트의 접속을 허용하는 메시지
S2C_DENY	클라이언트의 접속을 거부하는 메시지
S2C_USERLIST	접속자가 있거나 접속을 해제했을 때 생성된 공유자 목록을 전송하는 메시지

표 3. 클라이언트에서 서버로 전달되는 메시지
Table 3. Transmitted Messages from a Client to a Server

메시지	역할
C2S_CONNECT	회의도구 클라이언트가 서버에 접속을 요청하는 메시지
C2S_INITDATA	대형 디스플레이 장치의 JavaViewer가 회의 참여자의 Ultra VNC 서버에 접속하기 위하여 필요한 정보를 전송하는 메시지
CM2S_SORTSCRE	JavaViewer 창들을 바둑판식 배열하도록 요청하는 메시지
ENBYSQUARE	특정 JavaViewer 창의 자정한 크기를 조절하도록 요청하는 메시지
CM2S_RESIZESCRE	특정 JavaViewer 창의 자정한 크기를 조절하도록 요청하는 메시지
REEN	특정 JavaViewer 창을 가장 앞으로 위치하도록 요청하는 메시지
CM2S_ACTICEVI	특정 JavaViewer 창을 가장 앞으로 위치하도록 요청하는 메시지
EWER	접속해제를 알리는 메시지
C2S_CLOSE	접속해제를 알리는 메시지

CoDisplay 서버는 클라이언트의 요청 메시지를 전송받고 그 결과를 전송하는 송수신부와 전송받은 요청을 바탕으로 각 공유자의 화면을 제어하는 스크린 제어부, 공유자에 대한 정보를 관리하는 공유자 관리부로 구성된다. 또한 클라이언트로부터 전송된 화면을 수신하여 출력하기 위하여 내부적으로 JavaViewer를 사용하는 출력부로 구성된다. CoDisplay 클라이언트는 서버로부터 전송 받은 공유자 목록을 저장하는 클라이언트 데이터와 CoDisplay 서버에 제어 요청 메시지를 전송하고 그 결

과를 전송받는 제어부로 구성된다. 그리고 데스크톱 화면을 전송할 수 있도록 내부적으로 Ultra VNC 서버의 기능을 사용하는 송출부로 이루어진다. [그림 4]는 설계된 CoDisplay 서버와 클라이언트의 구조를 간단하게 나타내고 있다.

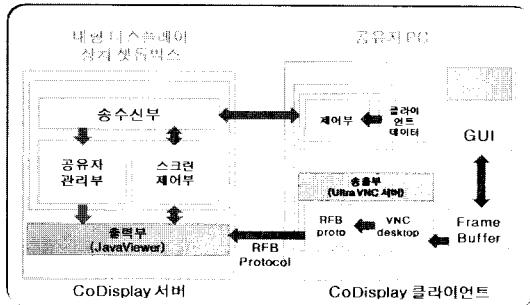


그림 4. CoDisplay의 구조

Fig. 4 An Architecture of CoDisplay

3.3. CoDisplay의 구현

CoDisplay 서버에서 각 공유자의 화면을 보여주는 창을 생성하고 생성된 창을 제어하기 위하여 JavaViewer 소프트웨어를 CoDisplay 서버의 클래스 형태로 포함한다. 또한 전송된 화면을 효과적으로 출력할 수 있도록 VncCanvas 클래스를 확장한다. [그림 5]는 JavaViewer를 포함하는 CoDisplay 서버가 공유자의 접속 요청을 처리하는 과정을 보인다.

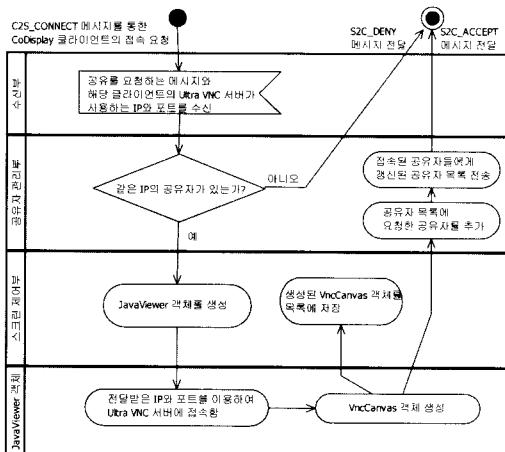


그림 5. 접속요청에 대한 CoDisplay 서버의 처리

Fig. 5 A process of CoDisplay server about a connection request

기존의 VncCanvas 클래스는 실행되는 시스템의 비디오 해상도가 출력하는 화면보다 낮은 경우 화면을 스크롤 하도록 구현되어 있다. 따라서 스크린 제어부에서 대형 디스플레이 장치의 비디오 해상도 정보를 얻고 생성될 화면의 넓이를 결정한다. 그리고 결정된 화면의 넓이에 따라 전체 화면을 표시할 수 있도록 VncCanvas 클래스를 확장하였다. VncCanvas 클래스는 FrameBuffer 형태로 Ultra VNC 서버에서 전송받은 화면 정보를 java.Image 객체 형태로 만들어 출력한다. 따라서 결정된 화면의 넓이에 따라 java.Image 객체의 넓이를 조절할 수 있도록 VncCanvas 클래스를 확장하여 서버로부터 전송된 화면의 전체를 출력할 수 있다.

java.Image의 넓이를 조절하는 방법을 이용하여 각 공유자들의 창을 바둑판식으로 배열하는 기능을 구현할 수 있다. 구현된 바둑판식 배열 기능은 스크린 제어부에서 처리되며 그 구현은 다음 [표 4]와 같다.

표 4. 바둑판식 배열 기능의 구현

Table 4. An implement of sort by square

```

/*각 창의 열 및 행의 개수 구함*/
count = VncCanvasList.size
rows = log2count
cols = log2count

/*각 창의 가로 및 세로 크기를 구함*/
Dimension = getVideoResolution()
width = Dimension.width / rows
height = Dimension.height / cols

/*각 VncCanvas 객체의 크기를 조절 및 배치*/
i = 0
x = 0, y = 0
while VncCanvasList.size > i do
    tempCV = VncCanvasList.getVncCanvas
    if i % cols = 0 then
        x = 0
        y = y + height
    end if
    tempCV.Frame.setLocation(x, y)
    tempCV.Image.drawImage(width ,height)
    x = x + width
    i++
end while

```

[그림 6]은 공유자가 4명일 때 바둑판식 배열의 결과를 보인다.

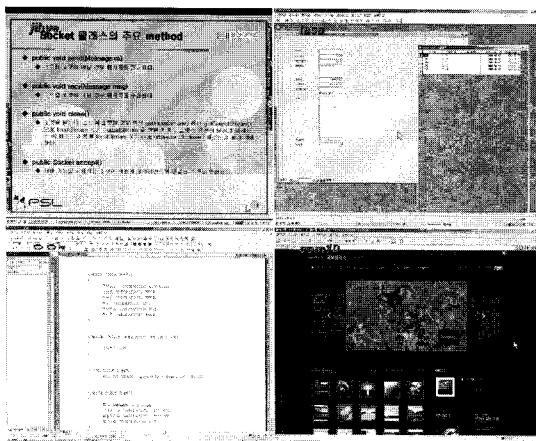


그림 6. 바둑판식 배열의 예
Fig. 6 An example of sort by square

특정 공유자의 화면을 선택하여 가장 앞으로 위치하도록 하는 기능은 자바의 윈도우 활성 및 비활성화 이벤트와 이벤트 리스너를 통하여 구현되었다.

IV. 결 론

본 논문에서는 네트워크를 통하여 대형 디스플레이 장치를 손쉽게 공유하여 의사전달을 효율적으로 진행 할 수 있는 회의도구인 CoDisplay의 개발에 대하여 기술하였다. 최근의 회의 참여자들은 효과적인 의견 및 의사 전달을 위하여 다양한 콘텐츠를 많이 사용하고 있다. 콘텐츠는 회의 장소에 마련된 대형 디스플레이 장치를 이용하여 출력되며 따라서 이를 여러 회의 참여자가 효과적으로 공유할 수 있는 수단이 필요하다. 그러나 유선 연결 및 USB와 같은 소형 저장장치를 이용한 대형 디스플레이 공유는 장치의 호환성 문제나 설치 및 철거의 문제, 출력 환경 구축의 비용문제 등의 어려운 점을 가지고 있다.

개발된 CoDisplay는 기존 대형 디스플레이 공유 방식이 가지는 문제점을 네트워크 연결을 통하여 해결한다. CoDisplay는 네트워크를 통하여 화면을 전송하고 출력하기 위하여 Ultra VNC 서버와 JavaViewer를 사용하며

공유자의 Ultra VNC 서버에 대형 디스플레이 장치와 연결된 PC의 JavaViewer가 접속할 수 있도록 한다. 그리고 빈번한 의사교환이 발생할 때, 공유된 사용자의 화면을 간편하게 제어할 수 있는 클라이언트 기반 인터페이스를 제공한다. 또한 CoDisplay는 자바 언어로 작성되어 다양한 플랫폼의 발표 환경을 효과적으로 지원할 수 있다.

현재 CoDisplay는 음성 데이터를 전송하고 출력하는 기능을 제공하지 않고 있다. 또한 CoDisplay 서버에 포함된 JavaViewer가 JVM을 반드시 필요로 하며 GUI 관련 API를 제공할 수 있는 실행환경을 필요로 하는 특성상 CoDisplay 서버를 동작하기 위하여 일반 PC급의 시스템이 반드시 필요하다. 따라서 향후연구를 통하여 음성 데이터를 전송할 수 있으며 임베디드 시스템으로 구성된 셋톱박스에서도 동작 가능한 CoDisplay 서버를 연구할 예정이다.

참고문헌

- [1] OpenOffice, “OpenOffice”,
<http://www.openoffice.org/>
- [2] Google Office, “Google Office”,
<http://doc.google.com/>
- [3] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, “Virtual Network Computing”, 1998
- [4] Thin Client Computing, “What is Thin Client Computing?”,
http://www.thinclient.net/whatis_thinclient.html
- [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] 이태호, 이홍창, 박양수, 이명준, “효과적인 협업을 위한 VNC의 확장”, 한국컴퓨터정보학회논문지, VOL.13 No.6, 2008. 12
- [10] Secure VNC, “Secure VNC”,
<http://www.laplink.com/products/lle/rc.asp>
- [11] 진성일, “Easy & Powerful e-Learning Content Authoring Tool”, KRnet 2005, 2005

저자소개



이태호(Taeho Lee)

2008.02 : 울산대학교 컴퓨터정보통신

공학부 졸업 (학사)

2008.03~현재 : 울산대학교 컴퓨터

정보통신공학과 석사과정

※ 관심분야 : 프로그래밍언어, 웹기반정보시스템, 공장
자동화시스템, 분산프로그래밍



박지혜(Ji-Hye Park)

2009.02 : 울산대학교 컴퓨터정보통신

공학부 졸업 (학사)

2009.03~현재 : 울산대학교 컴퓨터

정보통신공학과 석사과정

※ 관심분야 : 웹기반정보시스템, 생명정보학



이기훈(Ki-Hoon Lee)

2002.03~현재 : 울산대학교

컴퓨터정보통신공학부

재학

※ 관심분야 : 분산협업시스템



이명준(Myung-Joon Lee)

1980 : 서울대학교 수학과 졸업(학사)

1982 : 한국과학기술원 전산학과

졸업(석사)

1991 : 한국과학기술원 전산학과

졸업(박사)

1982 ~ 현재 : 울산대학교 컴퓨터정보통신공학부 (교수)

1993 ~ 1994 : 미국 버지니아대학 교환교수

2005 ~ 2006 : 미국 캘리포니아 주립대학 교환교수

※ 관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산
프로그래밍, 생물정보학, 센서네트워크 프로그래밍
환경