모바일 네트워크를 통한 프레젠테이션 지원 시스템 설계 및 구현

정승대°, 정찬용, 곽윤석, 권덕호, 정순기
경북대학교 컴퓨터공학과 가상현실연구실
sdjeong@vr.knu.ac.kr°, ccy400@hotmail.com, myshout@chol.com, kwondh24@dreamwiz.com, skjung@knu.ac.kr

Design and Implementation of Presentation Support System based on Mobile Networks

Seung Dae Jeong°, Chan Yong Jeong, Yoon Suk Kwak, Duk Ho Kwon, Soon Ki Jung
Dept. of Computer Engineering, Kyungpook National University, Daegu, South Korea

요 약

최근 무선 인터넷을 포함한 다양한 통신기술의 발달로 원격지에서 회의, 교육 및 발표를 위한 여러 가지 솔루션들이 마련되고 있으며, 그 이용영역이 확대되어가고 있는 추세이다. 그러나 대개의 시스템들은 영상이나 정보를 일방적으로 전달하는 방식이기 때문에 참여자가 수동적이 되기 쉬우며, 보다 능동적인 상호작용을 위해서는 참여자 각자에게 카메라와 같은 영상 장비가 양방향으로 설치되어야 하거나 다른 고가의 장비가 필요한 경우가 대부분이다. 본 논문에서는 이미 많은 사용자가 PDA나 Smart Phone을 보유하고 있고 점점 그 보급률이 향상되고 있다는 점에 착안하여, 이를 이용한 보다 능동적인 회의 참여 시스템을 제안하고, 각종 모바일 단말기와 PC간의 무선 통신을 통한 실시간 통신 기술을 바탕으로 하여 프레젠테이션 참여자의 상호작용을 지원하고 협동적인 회의가 가능하도록 하는 시스템을 구현하였다.

1. 서 론

한 장소에서 모여 의견을 논하는 전통적인 회의 방법에서 통계 자료와 미디어를 활용한 프레젠테이션 발표 형식, 그리고 인터넷을 포함한 다양한 이동 통신 기술을 이용한 원격 회의, 교육 및 발표에 이르기까지 서로의 의견과 정보를 교환하기 위한 회의 기술은 발전되어 왔으며, 여러 가지 방식의 솔루션들이 개발되어 왔다.

하지만 대개 단방향의 정보 전달과 지극히 제한적인 상호작용 방식으로 인해 회의 참여자가 일방적으로 정보를 받아들이는 수동적인 입장이 되기 쉽다. 이러한 단점을 보완하기 위해 보다 능동적인 상호작용을 지원하도록 개발된 솔루션들도 있지만 참여자 모두가 카메라와 같은 영상 장비가 양방향으로 설치되어 있거나 고가의 전용 단말기를 사용해야 한다는 단점이 있었다.[1,2,3]

본 연구에서는 별도의 추가 장비 없이 이미 참여자 각자가 보유하고 있는 모바일 기기를 통해 프레젠테이션에 참여하고 해당 기기의 능력에 맞는 다양한 방식의 상호작용을 통해 참여자의 보다 능동적인 의사 표현을 유도하는 시스템을 설계 구현하고자 한다.

2. 관련 연구

2.1 모바일 기기의 확산

최근 전세계적인 모바일 기기에 대한 연구와 지원 속에서 전자통신 기기에 대한 휴대성과 이동성이 중시되는 문화가 확산되어 개인 PC시장에서조차 데스크탑 PC에 비해 노트북 PC의 비율이 증가하고 있는 추세이다. 2005년도 국내 판매 비율은 PC와 노트북이 8:2 수준이지만 노트북의 비율이 꾸준히 증가하고 있는 추세이며, 미국과 유럽은 7:3, 일본은 5:5의 비율로 이러한 경향을 뒷받침 한다.

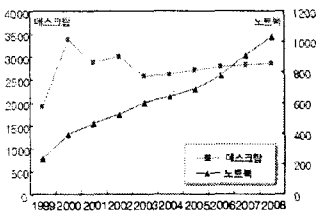

그림 1 국내 PC의 판매대수 추이 (천대)
전자정보센터(EIC), 2005

이러한 휴대성이 중요시되는 분위기는 앞으로도 계속 될 것이며, 2010년까지 전세계 성인 PC사용자의 75%가 PDA나 Smart Phone과 같은 차세대 PC[1]를 사용할 것으로 전망되고 있다.[4]

2.2 무선 인터넷이 정착단계로 진입

이동성이 부각된 무선 LAN과 휴대폰에 기반한 무선 인터넷 활용이 꾸준히 확대되고 있는 추세이다. 국내에서는 KT의 '네스팟(Netspot)'서비스가 출시 3년 6개월만에 50만 가입자를 돌파하였으며, 네스팟 존은 13,000여 개로 증가하였다. 해외에서는 핫스팟 존이 아태지역 29,400개, 미국 22,700개, 서유럽 25,000개 수준으로 언제든 무선 LAN에 접속할 수 있는 지역이 꾸준히 증가해 왔다.[4]

무선 인터넷의 이용률 또한 전체 인터넷 이용률에서 차지하는 비중이 꾸준히 증가하고 있으며, 이미 이동전화 기반의 무선 인터넷 서비스는 일상화되어 가고 서비스의 종류 또한 자료 검색에서 게임, 실시간 방송 등으로 다양화되었다.

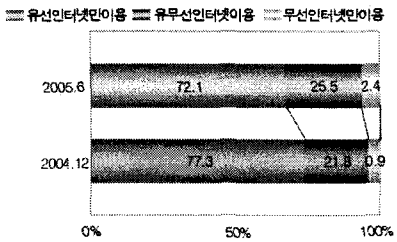

그림 2 유무선 인터넷 사용 비율
한국인터넷진흥원 (NIDA), 2005.7

2.3 Pebbles

PDA를 통해 PC의 화면을 공유하고 공동 작업을 하고자 하는 시도가 1997년부터 카네기 멜론 대학에서 연구되고 있다. PDA를 새로운 형태의 입출력 장치로 활용하여 PC의 능력을 확장시키고자 하는 연구로써 이에는 그룹화된 작업과 특화된 서버, 시리얼 통신, 적외선 통신과 같은 다양한 형태의 접속 방식도 포함되어 있다.[5]

2.4 PDA를 통한 PC의 확장

z2Remote2PC라는 제품은 PC의 화면을 PDA에서 보고 조정할 수 있는 상용 소프트웨어로써 화면의 회전 및 축소가 가능하며 마우스와 키보드 입력을 PC로 전송할 수 있어 PC상에서의 VNC(Virtual Network Computing)와 같은 기능을 수행하는 소프트웨어이다.[6]

또한 PDA를 별도의 모니터처럼 만들어 PC에서 멀티 모니터로 활용할 수 있는 상용 제품도 시판되고 있다.[7]

3. 프레젠테이션 지원 시스템 설계 및 구현

3.1 시스템의 구성

PC와 노트북은 펜티엄4 1GHz급, 256MB 이상의 메모리, 15MB이상의 저장 공간, Windows XP와 호환되는 OS를 최소 사양으로 초당 10프레임 이상의 화면을 목표로 하였으며, PDA는 PocketPC 2002와 호환되는 OS를 지닌 기기에서 초당 2.5프레임 이상의 화면을 목표로 하였다.[8]

(그림 3)에서와 같이 4가지의 어플리케이션으로 구성되며 각 어플리케이션의 역할은 다음과 같다.

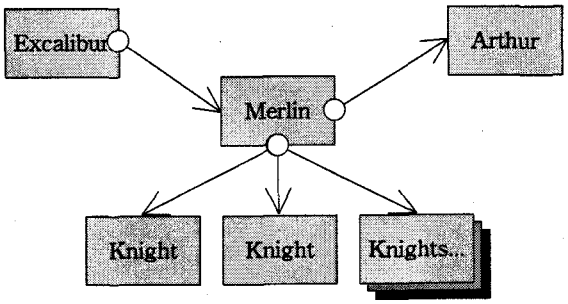

그림 3 Camelot 시스템

1. Excalibur
   - 실제 프레젠테이션이 실행되는 PC
   - 제어권을 가진 사용자가 원격으로 제어할 수 있다.
2. Merlin
   - 접속과 중간 변환 및 전송을 담당하는 서버 PC
   - 프로토콜이 플랫폼 독립적이게 구성되어 있어 각 플랫폼에 맞는 적절한 형태의 서버를 제작, 활용할 수 있도록 설계하였다.
3. Arthur
   - 발표자용 PC나 PDA에서 작동하는 클라이언트
   - 기본적으로 모든 통제권을 가지고 있으며, Kight가 제어권을 요청할 경우 허가 및 제어권 몰수를 할 수 있다.
4. Knight
   - 참가자용 PC나 PDA에서 작동하는 클라이언트
   - 프레젠테이션에 참가하여 공유된 화면을 전달받아 그려주며, 경우에 따라 제어권을 직접 요청해

1) 차세대 PC (Post-PC) : 휴대 및 착용이 가능한 소형·경량 정보기기로 웨어러블, 스마트폰, 핸드헬드 PC 등으로 분류

서 원격 PC를 조종할 수 있다.

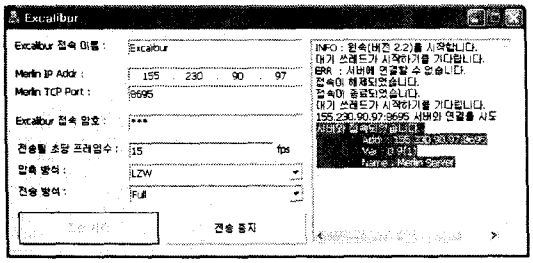

그림 4 Excalibur의 실행화면

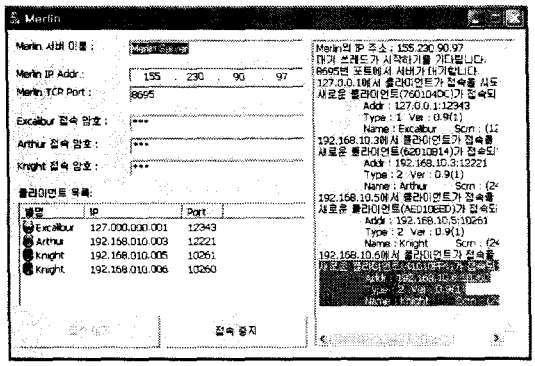

그림 5 Merlin의 실행화면

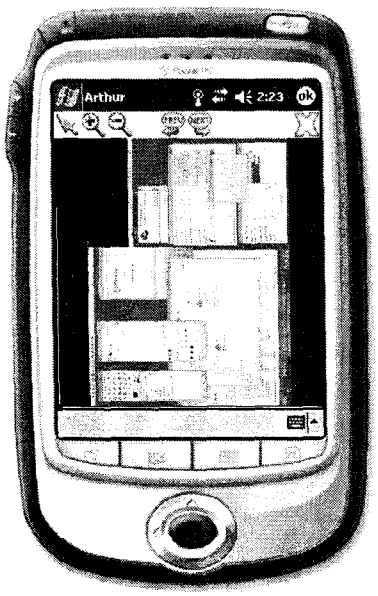

그림 6 PDA용 Arthur의 실행화면

3.2 상황별 처리 과정

작동 상황을 크게 4가지로 나누었으며, 각 상황의 처리과정을 미리 정하여 설계에 반영하였다.

3.2.1 사용자 접속

Merlin은 Excalibur, Arthur, Knight에 대해 각각 다른 3가지의 암호를 설정하게 된다. 클라이언트들은 자신을 나타낼 수 있는 별명을 설정하고 역할에 맞는 암호로 Merlin에 접속을 시도한다. Merlin은 접속 시도가 요청되면 클라이언트의 종류와 버전을 확인하고 암호를 비교하며, 이상이 없을 경우 클라이언트 기기의 상황을 전달받고 접속을 승인한다. 접속이 완료되면 클라이언트에서 원하는 화면을 요청하고 Merlin은 그에 맞게 변환된 화면을 전송하기 시작한다.

3.2.2 화면 전송

Excalibur는 실행 전 정해진 프레임대로 자신의 화면을 Merlin에게 전송한다. Merlin은 전송 받은 화면을 메인 화면으로 메모리에 보관한다. Arthur와 Knight는 Merlin에게 화면을 요청하고, 요청을 받은 Merlin은 해당 클라이언트에서 요청한 특성에 맞게 화면을 변환하고 전송한다.

3.2.3 제어권의 선택

기본적으로 모든 제어권은 Arthur가 가진다. Knight가 제어권을 요청할 경우 Arthur에게 승인할 것인지 묻는 메시지가 표시된다. Arthur는 허락 또는 거부할 수 있으며, 회의 진행을 방해하기 위해 계속된 요청에 거부할 수 있도록 제어권 요청을 잠글 수 있다. 제어권의 반납은 허가 없이 할 수 있으며, Arthur는 이미 수락한 제어권을 강제로 회수할 수 있다.

3.2.4 입력 이벤트의 전송

제어권을 가진 클라이언트가 마우스와 키보드 입력 이벤트를 Merlin에게 전송한다. Merlin은 수신된 이벤트가 정당한 제어권자에게서 나오는 것인지 판단한 후 Excalibur에게 전송한다. Excalibur는 전달받은 이벤트를 로컬에서 실행한다.

3.3 화면 처리

3.3.1 분할과 압축 모듈

화면 처리는 분할과 압축의 2가지의 모듈로 구성되어 있다. 압축 모듈은 전송할 화면의 비트 스트림을 다양한

압축 방식을 통하여 압축할 수 있도록 선택이 가능하게 분리된 모듈이며, 분할 모듈은 화면을 적절하게 분할하여 변화가 이루어진 최소 영역만을 검출할 수 있는 방식을 선택할 수 있도록 분리된 모듈이다.

3.3.2 차이영상 추출

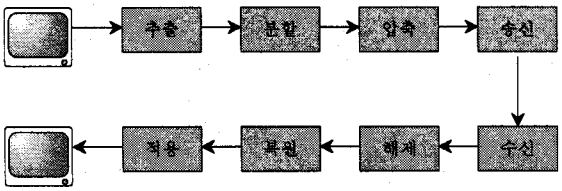

그림 7 화면 전송 절차 (Excalibur→Merlin, Merlin→Arthur,Knight)

(그림 7)에서 영상이 분할 모듈에 들어가기 전 추출 단계를 거치게 되는데, 이는 이전 영상과 다음 영상의 XOR 차이를 구하여 분할이 용이하도록 하고 압축 효율을 높이기 위한 단계이다.

(표 1)에서 C버퍼는 현재 영상이 담겨있는 버퍼이고 P버퍼는 이전 영상이 담겨있는 버퍼이다. 추출단계에서는 최종적으로 C버퍼에 P와 C를 XOR한 영상이 담기게 되며 이 영상은 변화가 없는 부분이 0으로 채워져 있다는 특징을 가지므로 분할 모듈에서 이를 기준으로 분할된 화면 중 전송해야 할 부분을 판단하게 된다. 또한 변화가 없는 화면일수록 0인 부분이 증가하므로 압축 모듈에서의 효율도 높아질 수 있다.

표 1 화면 송신전 차이영역 추출

Step	Buffer number		Process
	0	1	
1	Cx	Po	현재 화면을 C버퍼에 담는다.
2	Co	Po	C버퍼를 P버퍼에 XOR 시킨다.
3	Co	P+	버퍼 역할을 서로 교환한다.
4	Po	C+	C버퍼의 데이터를 송신한다.

화면 데이터를 수신한 후에는 압축을 해제하고 분할된 화면을 하나의 화면으로 복원한 후 해당 XOR화면과 이전의 화면을 결합하여 최종 화면을 만드는 과정을 수행하는데, 이는 (표 2)에 정리되어 있다.

원본 영상 한 장을 전송하는 방법에 비해 차이영상 방법은 버퍼도 2개나 가지며 처리 시간에 있어 추가적인 부하 시간이 발생하게 된다. 발생하는 부하 시간은 다음과 같다.

$$OverHead_1 = MemCopy(Size) + BitBlt(Size) + 1 \quad (1)$$

$$OverHead_2 = MemCopy(Size) + BitBlt(Size) + 2 \quad (2)$$

표 2 화면 수신후 차이영역 적용

Step	Buffer number		Process
	0	1	
1	Px	Co	버퍼 역할을 서로 교환한다.
2	Cx	Po	C버퍼에 데이터를 수신한다.
3	C+	Po	C버퍼를 P버퍼에 XOR 시킨다.
4	Cx	Po	버퍼 역할을 서로 교환한다.
5	Px	Co	C버퍼를 화면에 그린다.

송신할 때(1)와 수신할 때(2)의 차이는 버퍼의 역할을 바꾸는 연산 하나의 차이밖에 없으며, 비트블리트(BitBlt)의 경우 윈도우즈에서 가속 받을 수 있는 연산이므로 실제로 가장 큰 부하는 화면 영상을 버퍼로 옮기기 위해 수행하는 메모리 복사 연산이다. 그러나 송신할 정보를 최소화 시킬 수 있다는 점에서 아직은 상대적으로 속도가 느린 무선 통신을 사용한 전송을 위해 필수적인 부분이다.

3.3.3 영상의 크기 조정

상대적으로 메모리가 작은 모바일 기기에서 큰 화면 버퍼를 두 개나 유지한다는 것은 큰 부담일 수 있다. 그래서 모바일 기기의 화면 크기가 PC에 비해 크게 작다는 점에 착안해 기기의 능력에 따라 서버에서 화면의 크기를 조종한 후 전송할 수 있도록 Active Mode와 Passive Mode를 마련하였다.

Active Mode에서는 화면의 확대 축소와 스크롤을 클라이언트에서 조정하므로 서버는 전체 영상을 클라이언트로 전송하고 클라이언트는 많은 메모리를 사용하게 된다. 하지만 이 경우 전송이 지연되더라도 클라이언트에서 사용자의 입력에 대한 반응시간의 지연(latency)이 적다는 장점이 있다.

Passive Mode에서는 클라이언트가 확대 축소 비율과 스크롤 상태를 서버에게 전송하여 이미 처리된 화면을 수신하는 방식이다. 입력에 대해 다소 반응시간의 지연(latency)이 있으나 메모리를 적게 사용하고 CPU의 영향을 덜 받을 수 있다.

3.3.4 영상 버퍼

영상 처리에 사용되는 버퍼는 시스템의 가속을 받기위해 시스템에 의존한 장치의존적 비트맵(DDB, device dependent bitmap)로 되어 있다. 따라서 영상이 Excalibur와 Merlin 중 보다 낮은 색상의 색깊이(BPP,

bit per pixel)에 맞춰지게 된다. 분할 모듈까지 통과한 버퍼는 비로소 장치독립적 비트맵(DIB, device independent bitmap)에 복제되고 압축 모듈로 전달된다.

3.4 입력 이벤트 처리

키보드와 마우스를 대상으로 정보를 송신할 수 있도록 패킷이 구성되어 있으나 확장이 가능하도록 설계하여 차후에 다른 입력까지 전달 가능하도록 구현되었다.

현재 구현된 시스템은 윈도우즈 API를 사용하여 이벤트를 재생하지만 추후에 가상 HID(Human Interface Device) 드라이버를 제작하여 특수한 환경 하에서도 입력이 가능하도록 설계되었다.[9,10]

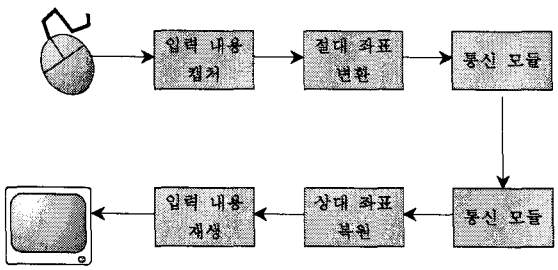

그림 8 입력 전송 절차

마우스나 기타 포인팅 장치의 경우 화면의 좌측상단에서 0으로 시작되는 절대 좌표로 변환한 후 Excalibur에서 자신에게 맞는 상대좌표로 복원하여 재생하게 된다.

3.5 소켓 통신

플랫폼에 독립적으로 작동할 수 있는 프로토콜을 구성하기 위해 정수는 네트워크 바이트 정렬(endian)에 맞게 정렬하여 전송하고, 모든 문자열은 UCS2로 인코딩하여 한글과 그 외의 언어를 사용하는데도 문제가 없도록 하였다.[11]

또한 소켓 스트림을 통해 전송하기 전 암호화 모듈을 계층적으로 삽입 가능하도록 설계하였으며 모바일 기기에 따로 탑재된 암호화 모듈이 있다면 이를 활용할 수 있도록 외부와 연동이 가능하게 구성하였다.

4. 결론 및 향후 연구

본 논문에서는 모바일 기기를 통하여 참여자가 필요에 따라 직접 프레젠테이션을 제어하고 상호작용 할 수 있는 프레젠테이션 지원 시스템을 설계하고 구현하였다.

이 시스템은 프레젠테이션용 클라이언트, 서버, 발표자 클라이언트, 참여자 클라이언트의 4가지 어플리케이션으로 구성되어 있으며, 각 클라이언트의 상태에 따라 적응적인 화면 공유 알고리즘을 적용하였다.

또한 별도의 장비 없이 참여자 각자가 이미 가지고 있는 모바일 기기를 통하여 참여할 수 있으므로 추가적인 비용이 발생하지 않으며, 플랫폼에 독립적이고 다양한 입력 방식을 추가할 수 있도록 설계되어 추후 다른 능력의 모바일 기기에서도 적용 가능하도록 확장이 용이하게 설계되었다.

향후 개별 참여자의 화면을 프레젠테이션 화면에 같이 포함한 채로 공유가 가능하도록 하여 발표자가 아닌 참여자가 가진 정보를 역 방향으로 공유할 수 있도록 확장하고, 발표장의 영상이나 음성을 같이 전송하여 참여자가 같은 회의장이 아닌 원격에서도 발표장의 분위기와 발표 내용을 전달 받을 수 있도록 연구를 진행시킬 계획이다.

참고문헌

[1] (주)컴버스 테크, http://www.combus.co.kr

[2] 한국폴리콤, http://www.kpcom.co.kr

[3] (주)와이즈파트너 UBizCenter, http://www.ubizcenter.co.kr

[4] 정명선, "통계로 본 2010년 유비쿼터스 사회 조망", 한국전산원 u-전략팀, 유비쿼터스사회연구시리즈 제5호, 2005.

[5] Brad A. Myers, "Using Handhelds and PCs together", Communicaions of the ACM, Vol. 44, No. 11, November 2001.

[6] z2 Software, http://www.z2software.com

[7] Innobec Technologies, http://www.innobec.com

[8] Douglas Boling, "Programming Microsoft Windows CE .NET 3rd Edition", Microsoft Press, 2003.

[9] Walter Oney, "Programming the Microsoft Windows Driver Model, 2nd Edition", Microsoft Press, 2002.

[10] 이봉석, "디바이스 드라이버 구조와 원리 그리고 제작 노하우", 가남사, 2004.

[11] The Unicode Consortium, http://www.unicode.org