

RTP 기반의 텔레포인팅 모듈의 구현

전재우°, 오삼권
호서대학교 컴퓨터공학부
e-mail: ohsk@office.hoseo.ac.kr

The Implementation of An RTP-Based Telepointing Module

Jaewoo Jeon°, Sam Kweon Oh
School of Computer Engineering, Hoseo University

요약

텔레포인팅은 지역 시스템의 공유윈도우에서 텔레포인터를 통해 발생하는 이벤트를 원격지 시스템의 공유윈도우에 동기적으로 표현하는 기법이다. 이러한 텔레포인팅은 다양한 CSCW분야에서 응용되고 있다. 본 논문은 다양한 텔레포인팅의 기능을 제공할 수 있고 응용프로그램에 상관없이 응용이 가능한 텔레포인팅 모듈을 설계하고 구현한다. 텔레포인팅 모듈은 텔레포인팅 이벤트의 송수신을 위해 RTP(Real-Time Transport Protocol)을 사용한다. 구현된 모듈이 잘 동작함을 보이기 위해 다가간의 텔레포인팅을 지원하는 화이트보드를 구현하고 인터넷 상에서 시연하였다.

1. 서론

CSCW시스템은 작업자들간의 공동작업을 효율적으로 진행하기 위해 다양한 의사소통기능과 공동작업인지(collaboration awareness) 기능들을 제공한다. 공동작업인지를 위한 기능들 중에 텔레포인팅은 지역 시스템의 공유윈도우에서 텔레포인터를 통해 발생한 이벤트를 원격지 시스템의 공유윈도우에 동기적으로 표현하는 기법이다. 여기서 텔레포인터란 공유윈도우 내에서 다른 사용자가 포인팅하고 있는 곳을 가리키는 작은 심벌(symbol)을 말한다[1].

본 논문은 다양한 텔레포인팅의 기능을 지원하고 응용프로그램의 종류에 상관없이 이용할 수 있는 RTP 기반의 텔레포인팅 모듈을 설계하고 구현한다. 텔레포인팅 응용프로그램들은 그 종류에 따라 텔레포인팅 기능요구사항이 다양할 수 있다. 예로 원격교육에서는 교육자의 포인터만을 실시간으로 표현하는 동적·단일 텔레포인팅을 제공할 수 있으며, 공동작업에서는 작업에 참여하고 있는 작업자들의 포인터들을 실시간으로 공유윈도우에 표현하는 동적·다중 텔레포인팅을 제공할 수 있다. 또한 화상회의에서는 설명하고자 하는 부분을 다수의 참여자들이 표시할 수 있도록 하는 정적·다중 텔레포인팅을 제공할 수 있다. 이를 위해 응용프로그램이 다양한 텔레포인팅 기능을 이용할 수 있도록 텔레포인팅 API

를 정의한다. 텔레포인팅 모듈은 텔레포인팅 메시지의 송수신을 위해 RTP(Real-Time Transport Protocol)[2]를 이용한다. 텔레포인팅 모듈에서 RTP를 이용하기 위해 RTP 통신모듈을 설계하고 구현한다. RTP 통신모듈은 텔레포인팅을 위한 RTP 패킷이 로드 형식에 따라 텔레포인팅 패킷을 정의하고 송수신 할 수 있도록 구현한다. 응용프로그램이 RTP를 사용함으로써 얻을 수 있는 장점은 RTP의 인터페이스를 통해 TCP, UDP, 또는 IPX등의 통신 프로토콜의 종류에 상관없이 데이터를 송수신할 수 있다는 것이다.

본 논문은 2장에서 텔레포인팅의 기능에 대해 설명하고, 3장에서 텔레포인팅 모듈을 설계한다. 4장에서는 텔레포인팅 모듈의 구현과 제공하는 API를 설명하고 마지막 5장에서 결론을 맺는다.

2. 텔레포인팅의 기능

텔레포인팅은 실시간 텔레포인터의 지원여부로 동적 또는 정적 텔레포인팅으로 구분될 수 있으며, 지원되는 텔레포인터의 개수에 따라 단일 또는 다중 텔레포인팅으로 구분될 수 있다. <표 1>은 텔레포인팅의 지원기능별 분류를 나타낸 것이다.

본 연구는 '99년도 정보통신부 대학기초연구지원을 받아 행해졌음.

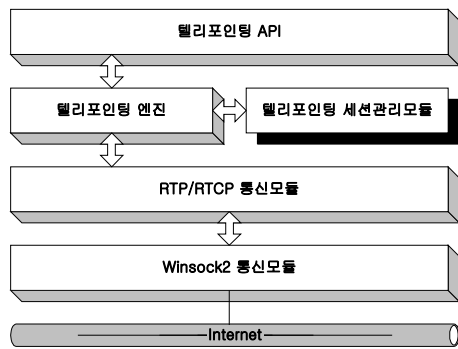
<표 1> 텔리포인팅의 기능별 분류

| 기능 | 분류 | 설명 |
|----------|----------|-------------------------------|
| 실시간 지원여부 | 동적 텔리포인팅 | 텔리포인팅의 움직임을 실시간으로 표현한다. |
| | 정적 텔리포인팅 | 작업자가 원할 때 특정 위치에 텔리포인터를 표현한다. |
| 텔리포인터의 수 | 단일 텔리포인팅 | 하나의 텔리포인터만을 표현한다. |
| | 다중 텔리포인팅 | 다수의 텔리포인터를 동시에 표현할 수 있다. |

3. 텔리포인팅 모듈의 설계

본 논문에서는 텔리포인팅 모듈의 데이터 전송을 위한 RTP의 하부프로토콜로써 TCP를 사용한다. TCP하에서의 멀티캐스팅은 현재 연결된 모든 참여자들에게 데이터를 전송하는 방식으로 수행된다. 신뢰성 있는 전송은 TCP가 연결지향형(connection-oriented) 프로토콜이므로 TCP자체에서 보장하고 있다. 그러나 실시간 전송을 위해 오디오/비디오와 같이 시간제약이 있는 실시간 전송 메커니즘은 본 논문에서 다루지 않으며 이벤트는 발생즉시 TCP를 통해 전송되는 것으로 한다.

(그림 1)은 텔리포인팅 모듈의 전체 구성을 나타내고 있다. 본 논문의 텔리포인팅 모듈은 윈도우 환경의 winsock2(windows socket version 2) 상에서 동작하며 TCP를 통해 텔리포인팅 메시지를 전송하도록 설계되었다. 텔리포인팅 모듈은 크게 winsock2 통신모듈, RTP/RTCP 통신모듈, 텔리포인팅 엔진, 텔리포인팅 세션관리모듈, 그리고 텔리포인팅 API로 구성된다.



(그림 1) 텔리포인팅 모듈의 구성

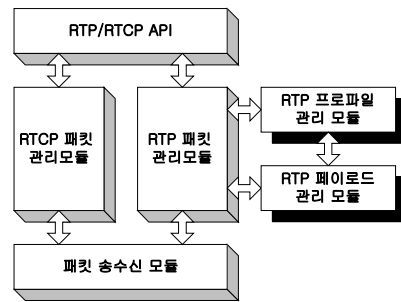
3.1. Winsock2 통신 모듈

Winsock2통신 모듈은 Microsoft 사의 윈도우 환경에서 제공되는 winsock2를 통해 데이터를 송수신하는 모듈이다. Winsock2 API들은 모두 C함수들로 제공된다. 따라서 RTP/RTCP 통신모듈에서 편리하게 이용할 수 있도록 winsock2 API를 클래스화하여 제공한다.

3.2. RTP/RTCP 통신모듈

RTP/RTCP 통신모듈은 텔리포인팅 엔진으로부터 데이터를 전달받아 이를 RTP 패킷으로 만들어 winsock2 모듈에게 전달하는 기능과 winsock2 모듈로부터 RTP 패킷을 수신하여 이를 텔리포인팅 엔

진으로 전달하는 기능을 수행한다. RTP/RTCP 통신 모듈의 구성은 (그림 2)와 같다.

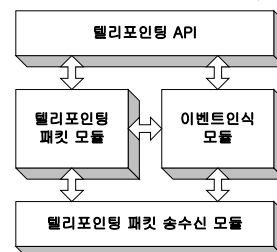


(그림 2) RTP/RTCP 통신모듈

- RTP 패킷 관리모듈: 텔리포인팅 엔진으로부터 데이터를 전달받아 이를 RTP 패킷으로 만들어 패킷 송수신모듈에게 전달하는 기능을 제공한다. 또한 패킷 송수신 모듈로부터 RTP 패킷을 전달받아 이를 텔리포인팅 엔진으로 전달하는 기능을 담당한다.
- RTCP 패킷 관리모듈: 텔리포인팅 엔진으로부터 데이터를 전달받아 이를 RTCP 패킷으로 만들어 패킷 송수신모듈에게 전달하는 기능을 제공한다. 또한 패킷 송수신 모듈로부터 RTCP 패킷을 전달받아 이를 텔리포인팅 엔진으로 전달하는 기능을 담당한다.
- RTP 프로파일 관리모듈: RTP 패킷 관리모듈에게 RTP를 통해 송수신 되는 데이터의 특성에 대한 정보를 제공하는 모듈이다.
- RTP 페이로드 관리모듈: RTP 패킷 관리모듈에게 RTP를 통해 전송되는 데이터의 종류에 대한 정보를 제공한다.
- 패킷 송수신 모듈: 패킷 송수신 모듈은 하위프로토콜을 통해 RTP 패킷을 송수신하는 기능을 제공한다.

3.3. 텔리포인팅 엔진

텔리포인팅 메시지를 송수신하는 기능을 제공한다. 응용프로그램으로부터 송신하고자 하는 메시지를 받아 RTP/RTCP 통신모듈에게 메시지를 전달하는 기능과 RTP/RTCP를 통해 수신된 텔리포인팅 메시지를 응용프로그램에게 전달하는 기능을 수행한다. 텔리포인팅 엔진의 구성은 (그림 3)과 같다.



(그림 3) 텔리포인팅 엔진

- 텔리포인팅 패킷 모듈: 응용프로그램으로부터 데이터를 전달받아 텔리포인팅 패킷 송수신 모듈로 전달하는 기능을 수행한다. 또한 텔리포인팅 메시지 송수신 모듈로부터 메시지를 전달받

아 응용프로그램에게 전달하는 기능을 수행한다.

- 이벤트 인식모듈: 이벤트 인식모듈은 응용프로그램으로부터 텔리포인팅 이벤트를 인식하는 기능을 수행한다.
- 텔리포인팅 메시지 송수신 모듈: RTP/RTCP 통신모듈을 통해 텔리포인팅 메시지를 송수신하는 기능을 수행한다.

텔리포인팅 메시지는 *이벤트 메시지*와 *제어 메시지*로 구분된다. 이벤트 메시지는 텔리포인터를 통해 발생한 이벤트의 전송을 위한 메시지이며, 제어 메시지는 세션 종료, 참여, 탈퇴, 그리고 배치조정 등을 위한 메시지이다. 이벤트 메시지는 RTP 패킷의 페이로드 부분을 사용하여 전송되고 제어 메시지는 RTCP 패킷 중 APP 패킷[2]을 사용하며 전송된다. <표 2>는 이벤트 메시지와 제어 메시지의 정보를 나타낸 것이다.

<표 2> 이벤트 메시지의 정보

| 메시지 | 정 보 | 크 기 |
|---------|--------------|-------|
| 이벤트 메시지 | X, Y 좌표 | 4 바이트 |
| | 아이콘 모양 | 4 바이트 |
| | 텔리포인터의 버튼 상태 | 4 바이트 |
| 제어 메시지 | 종류 | 4 바이트 |
| | 사용자 데이터의 길이 | 4 바이트 |
| | 사용자 데이터 | |

3.4. 텔리포인팅 세션 관리자

텔리포인팅 세션 관리자는 텔리포인팅 세션을 생성하거나 종료하는 기능을 제공한다. 또한 응용프로그램에게 한번에 여러 세션에 참여하여 작업할 수 있는 기능을 제공한다. 이를 위해 세션 관리자는 세션에 고유 ID값을 부여하여 텔리포인팅 메시지가 어떤 세션에서 발생한 것인지 알 수 있도록 한다. 부여된 ID값을 통해 어느 한 세션만 탈퇴하거나 종료할 수 있다.

4. 텔리포인팅 모듈의 구현

4.1. 구현환경 및 API

텔리포인팅 모듈은 Visual C++ 6.0을 이용하여 DLL(Dynamic Linking Library)의 형태로 구현되었다. TCP를 통해 텔리포인팅 메시지를 주고받도록 구현되었으며, 이를 위해 윈도우에서 제공하고 있는 winsock2 API를 이용하였다. winsock2 API를 이용한 이유는 winsock2 API가 Linux나 UNIX에서의 socket API와 문법적으로 거의 동일하기 때문에 포팅(porting)이 용이하기 때문이다.

<표3>은 구현된 텔리포인팅 모듈이 제공하는 텔리포인팅 API를 요약한 것이다. tpSessionBegin API는 텔리포인팅 세션을 생성하는 기능을 수행한다. 텔리포인팅 세션이 시작되면 세션 ID값이 리턴된다. tpSessionEnd는 생성된 텔리포인팅 세션을 종료하는 API이다. 이때 전달인자로 생성된 세션의 ID를 전달한다. tpSessionJoin는 텔리포인팅 세션에 참

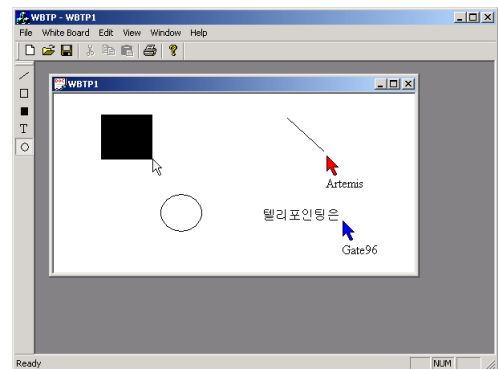
여하는 기능을 수행하는 API이며, tpSessionLeave는 참여중인 세션에서 탈퇴하는 기능을 제공하는 API이다. tpSend와 tpReceive는 텔리포인팅 메시지를 송수신하는 API이며, tpSendAck와 tpSendNack는 각각 ACK메시지와 NACK메시지를 송신하는 API이다. 또한 tpCallback은 텔리포인팅 시에 발생하는 여러 가지 이벤트를 응용프로그램에게 알리는 API이다. tpLayoutNegotiation은 세션의 책임자와 응용프로그램의 배치를 협의하기 위한 API이며, 마지막으로 tpLayoutAdjust는 텔리포인팅 중에 응용프로그램의 배치가 변경되는 경우 각 참여자들의 배치를 조정하기 위한 API이다.

<표 3> 텔리포인팅 API

| API | 전달인자 | 리턴값 |
|---------------------|---------------|--------------------|
| tpSessionBegin | 포트번호. | 세션의 ID. |
| tpSessionEnd | 세션 ID. | 없음. |
| tpSessionJoin | IP주소와 포트번호. | 세션의 ID. |
| tpSessionLeave | 세션 ID. | 없음. |
| tpLayoutNegotiation | 배치정보. | TRUE 또는 FALSE. |
| tpLayoutAdjust | 배치정보. | 없음. |
| tpSend | 송신할 메시지의 포인터. | TRUE 또는 FALSE. |
| tpSendAck | 없음. | TRUE 또는 FALSE. |
| tpSendNack | 없음. | TRUE 또는 FALSE. |
| tpReceive | 수신할 메시지의 포인터. | 수신된 메시지의 바이트단위 길이. |
| tpCallback | 이벤트 정보. | 없음. |

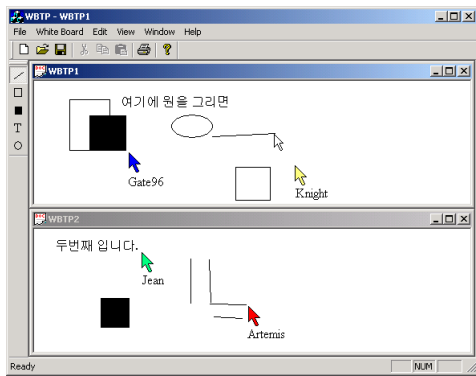
4.2. 구현사례 - 텔리포인팅을 이용한 화이트보드

구현된 텔리포인팅 모듈이 잘 동작함을 보이기 위해 다자간의 텔리포인팅이 가능한 화이트 보드를 구현하고 10Mbps 급의 LAN환경에서 시연하였다. 구현환경으로는 펜티엄II 350Mhz 급의 PC에서 Microsoft 사의 윈도우 98 SE(Second Edition) 운영체제를 사용하였으며, 개발도구로는 Visual C++ 6.0을 이용하였다. 시연환경으로 펜티엄급의 PC를 클라이언트로 사용하였다. (그림 4)와 (그림 5)는 화이트 보드가 실행된 모습이다.



(그림 4) 단일 세션에서의 텔리포인팅

(그림 4)는 하나의 세션에서 3명의 참여자가 작업을 하고 있는 모습이다. 흰색으로 표시된 포인터가 세션책임자의 포인터이다. 다른 참여자의 텔리포인터는 참여자 별로 색이 다르게 표시되었으며 참여자의 ID가 텔리포인터 밑에 표시되고 있다. (그림 5)는 두 개의 세션에 동시에 참여하고 작업중인 모습이다. (그림 4)와 마찬가지로 서로 다른 색으로 참여자들의 텔리포인터를 표시하고 있고 참여자의 ID가 포인터 하단에 표시되고 있다. 최대 5명의 작업자가 참여하여 화이트 보드를 사용하였으며, 동시에 두 개의 세션을 열어 작업해 보았다. 시연의 결과로, 텔리포인터의 움직임이 약간 부드럽지는 않았으나 무리 없이 작업을 진행할 수 있었다.



(그림 5) 다중 세션에서의 텔리포인터

5. 결론

본 논문에서는 응용프로그램의 종류에 상관없이 사용할 수 있고, 다양한 텔리포인터의 기능을 제공하는 RTP 기반의 텔리포인터 모듈을 설계하고 구현하였다. 텔리포인터 메시지는 텔리포인터를 위한 RTP 페이로드 형식에 따라 구조체를 정의하고 RTP를 통해 송수신 될 수 있도록 하고 이를 위한 RTP 통신모듈을 설계하고 구현하였다. 구현된 RTP 통신모듈은 하부의 통신 프로토콜로써 TCP를 사용한다. 구현된 텔리포인터 모듈이 잘 동작함을 확인하기 위해 윈도우 운영체제에서 동작하는 화이트 보드를 구현하였다. LAN환경에서 5명의 작업자가 참여한 후 시연한 결과 텔리포인터가 잘 수행됨을 확인하였다.

참고문헌

[1] Tom Brink, "CSCW & Groupware Index", <http://www.usabilityfirst.com/cscw/>. Interaction, Hamilton, New Zealand, 24-27, Nov., 1996.

[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan., 1996.

[3] Andrew S. Tannenbaum, "Computer Networks 3rd Edition", Prentice-Hall, 1996.

[4] C. Kenneth Miller, "Multicast Networking and Application", Addison Wesley Longman Inc., 1999.

[5] Eve M. Schooler, "Conferencing and Collaborative Computing", Multimedia Systems, Springer-Verlag, Oct., 1996.

[6] Jonathan Grudin, "Computer-Supported Cooperative Work: History and Focus", IEEE Computer, Vol.27, No.5, pp.19-25, May, 1994.

[7] 전재우, 오삼권, "동기적 공동저작 및 프리젠테이션을 위한 텔리포인터", 추계학술발표논문집, 한국정보과학회, 제25권, 제2호, pp.279-281, 10월, 1998.

[8] Saul Greenberg, Carl Gutwin, Mark Roseman, "Semantic Telepointer for Groupware", Proceedings of OzCHI'96 6th Australian Conference on Computer-Human

[9] Walter Reinhard, *et al*, "CSCW Tools: Concepts and Architectures", IEEE Computer, Vol.27, No.5, pp.28-36, May, 1994.