

86-89

Internet网

Intranet网

CSCW'

白板系统

21

计算机科学 1999 Vol. 26 No. 2

Internet/Intranet 上 CSCW 中白板系统的设计与实现^{*}

The Design & Implementation of White Board System in CSCW over Internet/Intranet

王魁生 李人厚 倪小东

TP393

(西安交通大学科学与工程研究院系统工程研究所 西安 710049)

Abstract Along with the development of Internet/Intranet technology, the White-Board is a very useful cooperative tool in CSCW system. This paper introduces the basic concept of CSCW, then discusses the model of White-Board system, and on the base of TCP/IP protocol, presents the constitute of White-Board system and some key techniques in the implementation of White-Board system with the Microsoft Visual C++ development environment.

Keywords CSCW, White-Board, TCP/IP protocol, Socket, Thread process, OLE component

一、前言

计算机支持的协同工作(CSCW—Computer Supported Cooperative Work)是一门由计算机科学、系统工程、心理学和社会学等多个学科综合而成的新的交叉学科。CSCW为在时空上分散的人们提供了一个“面对面”和“你见即我见”WYSIWIS的协同工作环境,支持多个时间上分离、空间上分布而工作又相互依赖的协作成员的协同工作,使计算机系统从传统的只能提高个体工作效率变为能提高群体工作效率,因而受到日益广泛的重视。CSCW技术主要是研究如何利用计算机技术、多媒体技术和网络通信技术,支持工作群体成员在共享环境下的协同工作、交互协商、分工合作,共同完成某些任务。它由对群体工作方式的理解决和支持群体工作方式的相技术两部分组成,CSCW突破了传统的计算机孤立应用的概念,将计算机技术、多媒体技术、网络通信技术以及各种社会科学紧密联系在一起,向人们提供了一种全新的交流与工作模式,具有广泛的应用前景。CSCW使计算机系统从传统的只能提高个体工作效率变为能提高群体工作效率,并展示了巨大的应用潜力。

Internet已经热遍全球,随之而来,在企业界掀起更大风波的是Intranet。Intranet即企业内部网,是企业级的Internet。它采用Internet技术,利用

TCP/IP协议、Web服务器及浏览器,主要实现企业内部以及外部的信息交流,而且Intranet不是从建立起来的空中楼阁,它是在企业现有的网络硬件、软件和服务器的基础上,采用Internet技术和标准,通过提供多种Intranet服务来完成的。国际著名咨询公司Forrester的报告预测,到2000年,Intranet的业务增长势头将远远高于Internet,Intranet在近年的发展趋势是将取代专用网络操作系统。

CSCW技术支持多个时间上分离、空间上分布而工作又相互依赖的协作成员的协同工作;另一方面,与Internet相比,Intranet除了信息交流更加安全可靠之外,还有一个最大的优势就是它的传输速率,这为多媒体信息的实时传输提供了比较优越的信道。所以,Intranet为CSCW的应用和研究提供了一个更加广泛和先进的环境。本文主要论述如何在Internet/Intranet环境下,利用TCP/IP协议来设计和开发CSCW领域中最基本的协同式工作工具——白板系统的设计与实现。

二、白板的信息模型

所谓白板(Whiteboard),指的是一个虚拟公用区域,在CSCW中参加协作的各个成员的计算机上都有一个对白板的视图,白板上的内容既可以是随手画的简单图形(直线、曲线、矩形等)、文字、图像,也可以是任何其他类型的数据,如协同设计中的各

* 本项目受国家自然科学基金资助

种模型,只要这些数据类型可以被白板识别印可,白板是共享数据,当一个用户修改了其中的数据之后,就要把修改立即通知给其他用户,让他们及时更新各自在白板上的视图,就是说,当某一个参加者在自己的视图上面添加、删除、移动、修改对象后,其结果应该立即反映在其他所有参加者的视图上,即所谓的“你见即我见”(WYSIWIS)。

白板系统的实质是数据通信,要求有可靠的传输机制和通信模型,在具体实现时,目前有两种组成方式:对等方式(Peer-to-Peer)和客户机/服务器方式(Client/Server)。

使用对等方式实现时,信息采用(n)的方式进行保存和处理(n为使用白板系统的用户的个数),也就是说,各个站点的功能是相同的,每个站点都保留一份共享数据的拷贝,一个站点的数据修改后就通知其它所有站点进行同样的修改操作。这对于两三个客户之间的协同工作还是比较方便的,因为它不需要单独的服务器;但是要在 Internet/Intranet 网络上实现多个客户之间的协同工作时,这种模式就会出现一个十分关键的问题——难以维持共享数据的一致性,要解决它就非常麻烦。

使用客户机/服务器方式实现时,信息采用(n+1)的方式进行保存和处理(n为使用白板系统的用户数),也就是说,它是由一个服务器和若干个客户机组成,服务器维护一份全局数据(即:1),这也是维护数据一致性的依据,后加入的客户可以从服务器这里得到当前的系统信息;而每个客户机上都有一份全局数据的拷贝(即:n),用户在客户机上对数据进行操作,某个用户在它的客户机上进行一次修改操作后,客户机并不立即更新本地数据,而是先把修改请求传递给服务器,服务器更新全局数据后再向所有客户机发出修改通知,各客户机接到修改通知后才更新本地数据。

我们在具体实现时,白板系统采用典型的 Client/Server 结构模型,在通信过程中,白板信息采用(n+1)的方式进行保存和处理(n为使用电子白板的用户数),其结构如图1所示。

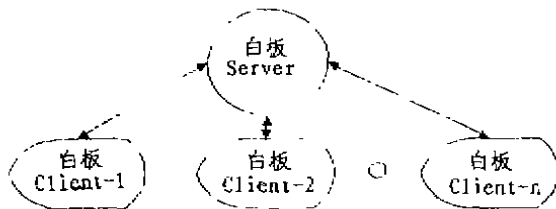


图1 白板系统中的信息模型

采用(n+1)的信息结构的优点在于:(1)当一个客户在自己的白板视图上产生了一个对象时,系统只须将该信息发送给白板 Server,由白板 Server 负责将该信息发送给所有使用白板的客户,这样就减轻了客户机的负担;(2)当一个客户修改白板视图上的一个对象时,白板 Server 便可及时地对所有客户端的该对象建立冲突监测机制;(3)当有一个新客户要加入时,白板 Server 会为该用户提供其他客户已经建立的所有对象,使该客户与其他人能够同步使用白板。

三、白板的总体结构

白板系统由白板 Server 和白板 Client 两大部分组成,首先由白板 Server 负责白板系统的集中式服务器。开始,该服务器建立一个监听白板 Client 请求连接的 Socket,当有连接请求时,它便确认客户的连接请求,并在白板 Server 上建立一个相应的服务对象——客户 Socket,同时建立该服务对象的链表队列,然后向该对象发送目前白板系统的有关信息(包括:使用该白板系统的成员列表和已启用白板的成员名称),同时还发送白板系统当前已经建立的相关对象,最后启动该服务对象的异步监听功能,以便能够及时地响应该客户的服务请求。

当参加协作的用户启用白板进行交流时,首先由白板 Client 建立自己的 Socket 对象,并向系统的白板 Server 发出一个连接请求,当接到白板 Server 的确认信号之后便会为客户建立白板系统所需的发送和接收缓冲区及其有关工具(包括:设置线型工具、设置颜色工具、绘图工具、编辑工具等),同时建立并启动一个用于接收和处理白板系统信息的信息处理线程,由该线程在后台自主地处理所接收的信息;同时前台的程序负责处理客户对白板系统的操作。当参加协作的用户要退出白板系统时,白板 Client 先向白板 Server 发出断开连接的消息,然后就终止消息处理线程的运行,撤消自己的 Socket 对象,最后释放白板 Client 所占用的资源,返回 Net-Comp 系统。

当白板 Server 接收到一个白板 Client 发来的断开连接的消息时,它首先从服务对象的链表队列中删除该对象,同时撤消该客户的 Socket 对象,最后白板 Server 还要判别服务对象的链表队列是否为空,如果为空,则表示所有客户都已经退出了白板系统,那么白板 Server 就需要清除所保存的所有消息并释放消息所占用的系统空间,重新等待新的连

接。白板系统的总体结构如图 2 所示。

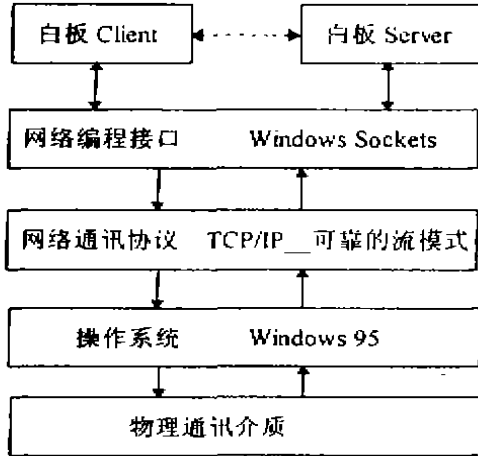


图 2 白板系统的总体结构

四、实现白板系统的关键技术

1. 采用可靠模式的 TCP/IP 通信协议

为了面向 Internet/Intranet 网络,跨越远程客户,白板子系统采用了 TCP/IP 通信协议,由于白板子系统属于数字信息,为了可靠传输,所以采用可靠的面向连接的 Stream 模式的 TCP/IP 协议,使用 Microsoft Visual C++ 5.0 提供的 WinSock 接口实现,其通信模型如图 3 所示。

2. 多线程技术

白板系统一方面要实时地响应用户的交互操作,另一方面还要及时地处理网络信息,所以需要采用 Windows95 或者 Windows NT 提供的多线程技术来实现并行处理。在白板 Server 上,首先由主程序自动产生一个主线程,然后调用 Csocket 类的实例 CListenServer,系统自动产生一个负责监听网络上白板 Client 请求连接的线程,当接收到一个连接请求后,再由主线程调用 Csocket 类的实例 CConnectServer,系统便会自动产生一个负责监听这个网络连接上的线程,来响应这个白板 Client 的网络信息。退出时,由主线程逐个删除 Csocket 类的实例,系统便会自动终止相应的线程;而在白板 Client 上,首先由主程序自动产生一个主线程,然后调用 Csocket 类的实例 CMsgSocket,系统自动产生一个负责网络通信的线程,同时调用 CWinThread 类的实例 CMsgProcessThread,由应用程序自生成一个处理网络信息的线程(通过调用 CreateThread()来完成),负责将接收到的信息进行分析 and 处理。

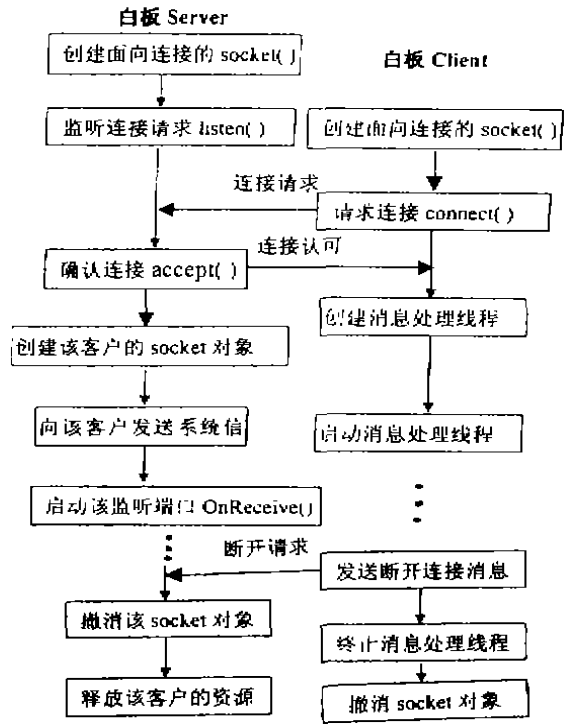


图 3 白板系统的通信模型

而主线程同时还要负责响应用户通过键盘、鼠标和显示器的信息交互操作,还可以通过调用 SetThreadPriority()来设置线程的优先级,以使整个系统得到合理的调度,最后,退出系统时,首先要终止应用程序自生成的线程,然后删除 Csocket 类的实例以便终止网络通信的线程,最后才是退出主线程,关闭这个应用程序。

3. 外部接口——OLE Automation 构件技术

白板系统是 CSCW 应用系统中的一个协同工作的工具,为了与其他子系统有机地集成起来,所以采用了 OLE Automation 构件技术。OLE (Object Linking and Embedding) 是有关对象互操作和集成的技术,在 COM (Component Object Model) 的基础上,一个 OLE 对象提供若干个接口,客户程序可以通过接口指针来访问其成员函数。在 Microsoft VC++ 中提供了 OLE Automation 构件技术,使得 OLE 对象更易应用,具体实现时,被编程的对象(如白板系统)被称为 Automation 对象,可能通过 Automation 对象的属性和方法来提供向外的接口,在白板系统中提供的外部接口如图 4 和图 5 所示。

这些方法被 Microsoft VC++ 自动封装在扩展名为 TLB 的文件之中,当其它应用系统需要使用白