

计算机会议系统

CSCW

Internet网

15

电子信件

计算机科学1999Vol. 26No. 7

桌面文档会议系统实现中的关键技术分析*)

63-66,73

Key Techniques for Desktop Conferencing System Implementation

李成锴 茅兵 陈道善 谢立

TP393

(南京大学计算机科学与技术系 计算机软件新技术国家重点实验室 南京210093)

Abstract In this paper, we introduce the implementation issues of desktop conferencing system. Emphasis is given to the analysis of key techniques involving architecture, access control, awareness, role management, concurrency control, latecomer and synchronous.

Keywords CSCW, Groupware, Awareness, Concurrency control, Whiteboard

1 引言

随着工作站和个人计算机性能的飞速提高以及计算机网络和分布式计算技术的发展,特别是Internet应用的迅速普及,使得CSCW的研究成为当前的热点,同时,合作与交流在当今的社会生活中越来越普遍,并成为大型项目工程必不可少的工作方式,这也是CSCW技术的发展动力。CSCW反映了人们对计算机的功能需求的改变,即希望计算机系统的功能从传统地解决计算问题发展为辅助多个用户之间的合作活动^[1],目前,CSCW技术已经应用到了各个领域,出现了不少实验性和实用的系统,例如桌面会议系统、协同写作系统、远程教育系统、虚拟实验室等等,不胜枚举,CSCW系统也称为协同系统,通常按照用户的交互模式和地理位置对协同系统进行分类,多个用户之间的交互模式可以是同步的或者异步的;用户的地理位置可以是在同一地点或在不同的地方。这种时空分类的方法可以将协同系统归结为以下四种模式^[2]:同步交互模式,例如会议室系统;同步分布式模式,例如远程视频会议系统、协同写作、共享白板;异步模式,例如留言板;异步分布式模式,例如电子邮件、电子公告板系统等。

认识到各种系统模式的区别对于协同系统的设计和开发是很重要的,因为它们的网络协议、存储需求等等各不相同。桌面文档会议系统属于同步分布

式模式,这种模式的系统的协同需求是最典型和最复杂的。本文从体系结构、访问控制、协同感知、角色管理,并发控制、中途加入、同步等几个方面,详细分析了桌面文档会议系统实现中的关键技术。

2 桌面文档会议系统的功能部件

桌面文档会议系统的目的是提供一个虚拟会议环境,使地理上分布的用户通过网络就能够召开会议,共同讨论问题、共享信息、协同工作,这种会议方式具有交互性、分布性、实时性、协作性等特点,提高了工作效率。一般桌面文档会议系统中应该提供以下功能部件:

共享白板:日常生活中,黑板在人们的会议和讨论中经常被使用,便于人们发表和阐述自己的观点,互相交流,提高工作效率。共享白板模拟黑板这一工具,并且结合图形、图像、语音、视频等功能,具有比黑板更丰富的表现力。

协同编辑器:或称为协同写作、合著系统,提供协同编写会议文档、报告的功能,支持多个编写者同时编辑一篇文档。

聊天室:与会者可以在聊天室中以文本方式讨论问题、聊天。

视频工具:这一工具可以把摄像头摄取的用户视频信息显示给其他与会者。

语音工具:可以实时传输与会者的语音信息,与

*)本文得到国家“863”高技术计划项目(863-306-02-07)基金资助,李成锴 硕士研究生,主要研究方向为分布式计算、CSCW,茅兵 博士、副教授,主要研究方向为分布式计算、CSCW,陈道善 教授,主要研究方向为分布/并行系统,谢立 教授,博士生导师,主要研究方向为分布/并行系统。

视频工具一起增加临场感,创造了协同系统需要的共同工作的虚拟环境。

表决器:提供支持与会者对有关问题进行表决的工具。

3 桌面文档会议系统实现中的关键技术

CSCW 系统的研究涉及计算机科学、认知科学、心理学、社会学、行为科学等诸多领域。在计算机科学中,涉及 CSCW 系统的领域主要包括分布式系统、网络和通信技术、人机交互技术、多媒体技术、人工智能等等。下面从体系结构、访问控制、协同感知、角色管理、并发控制、中途加入、同步等几个方面出发,讨论实现桌面文档会议系统的关键技术。

体系结构:CSCW 系统一般可以采用集中式、复制式和混合式三种结构^[3-4]。集中式结构中,应用程序运行在一个中心节点上,其它节点提供与用户交互的界面。由中心节点进行输入复用和输出过滤(multiplex/demultiplex),用户的输入都要发送到该节点,经应用程序处理后再把输出信息转发给所有用户。采用这种结构,所有的操作都是序列化的,控制简单,易于实现一致性。但是由于中心节点既要负责应用程序的运行,又要负责信息转发,将成为整个系统的瓶颈,造成效率低下,而且健壮性差。复制式结构中,所有的用户方都有应用程序的副本在运行,它处理用户的输入,产生本地操作结果,并把结果发送到其他用户,这样系统中没有负荷特别大的服务器,系统性能得以保证,且坚定性好,在这种结构中,由于分布式系统中各节点的处理能力和网络速度与时延的差别,会造成操作的不一致,需要进行并发控制和冲突消解,但是各个结点是对等的不能进行集中控制,因此控制复杂。混合式结构结合集中式和复制式的优点,比较灵活。在桌面文档会议系统中,系统的执行效率对于系统是否实用至关重要,效率低下反而会阻碍工作的进展,而数据的一致性也是必须保证的,因此适宜采用混合结构。一种可能的系统体系结构如图1所示。应用程序的执行是分布在各个用户的机器上的,而数据和控制在服务器上。本地的应用程序接收到用户的输入事件后,把事件发送到服务器,服务器根据各个客户方发送来的信息,进行适当的控制,然后向各个客户方发送操作序列,再由各个客户方的应用程序执行相应的操作。

访问控制:协同系统中的用户需要对数据进行访问,然而谁能够以何种方式来访问什么数据是需要规定的^[1]。数据安全性和可访问性的要求决定了

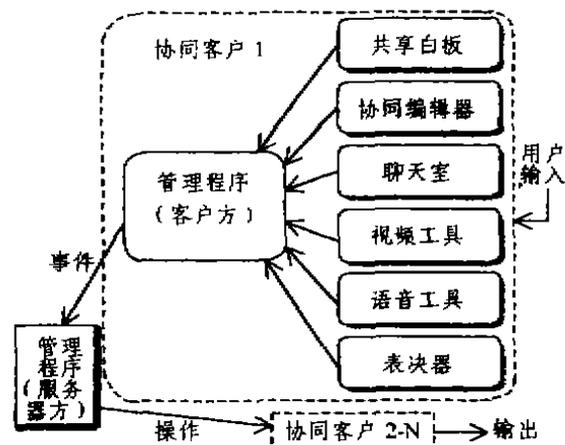


图1 混合式结构

必须制定数据共享的策略,有的数据可以被工作组中所有的用户共享访问,而有的数据是属于私有信息,不能够共享。在桌面会议系统中,不仅要提供共享白板和协同编辑器,还应为每个用户提供私有白板和私有编辑器。私有白板和私有编辑器的信息是不共享的,其他用户不能获取。它们可以让用户在进行问题的初步考虑时使用,使得在共享白板和协同编辑器上进行操作时有的放矢。这种既支持共享、又保护用户私有数据的方法可以提高工作效率,减少无用信息的传输,并可以增加用户的信心,特别是对不熟练的用户。

协同感知(awareness):在一般分布式系统中,用户应该感觉不到互相的存在,即保持透明性。与此不同的是,要协同就必须感知,CSCW 系统要使用户的行为能够对整个协同工作产生影响,就要让用户的行为被相互感知,并以显示的方式体现在协同工作的成果上。协同感知技术是桌面文档会议系统中非常重要的技术,有关的内容包括感知范围、感知模式、感知粒度的制定,以决定用户的某种行为能够以何种方式被其他用户感知^[1]。

(1)感知范围:桌面会议系统需要针对不同的情况来制定感知范围,这意味着并不是用户的所有行为都被感知,因为有些行为只对用户自己的视图起作用,有些行为是秘密性质的,有些是没有必要感知的。例如,用户可以选择字体的大小和颜色,在白板上输入字符,使自己输入的内容区别于别的用户,但是这种对字体的选择并不影响别人输入的字符的外观。又例如用户进行两方秘密会谈时,谈话的内容不应该公开。

(2)感知模式:各个用户的屏幕可以用不同的布局方式来表示相同的内容,用户各自的界面可以包含同一元素的不同部分,另外应用程序本身可以支持不同的显示方式从而由用户自己来定制,例如不同的颜色、鼠标形状^[5]。一般可以有三种感知模式。屏幕共享感知模式使协同用户的屏幕具有相同的外观,这种模式要求同构的系统;窗口共享感知模式使感知信息在协同用户的相应窗口以相同的方式显示,同构系统和异构系统都可以支持这种模式;对象共享感知模式使同样的感知信息以不同的方式表达,在这种模式下,某个用户窗口中的一段文本可以与另一用户的窗口中的一幅图像表达同样的内容。显然对象共享感知模式适合于异构系统。

(3)感知粒度:协同系统可以支持多种粒度的感知。细粒度的感知是在动作级上的感知,中粒度的感知是在过程级上的,而粗粒度的感知是在对象级上的。对于白板上的内容可以采用细粒度感知,例如某用户绘制一个框图供全组讨论,是以逐条线段出现的,可以让其他人清楚地看到全过程,如果一个图形全部绘制好以后再显示在其他用户白板上,可能会造成莫名其妙感觉,也可能会分散注意力。而聊天室的内容是逐句出现的,这种较粗的感知粒度已经足够。在协同编辑器中,可以选取不同的感知粒度,这样一个用户所输入的文本以不同的方式显示在其他用户的屏幕上。细粒度感知使文本逐字(包括退格键)出现,中粒度感知使文本逐段出现(由回车键的输入触发),粗粒度感知使文本逐章节出现(由一个编辑阶段的结束触发)。

角色管理:在 CSCW 系统中,多个用户通过共同的任务和共享的工作环境而结合在一起,称为一个组(group),因此 CSCW 系统也被称为群件(groupware)。各个协同用户的权力与责任是不同的,用户的角色是赋予用户的权限的集合,这些权限

可以包括对数据的访问控制和对操作的执行控制。进行角色管理有三个方面的作用:首先可以简化权限的管理,不需要对每一个用户指明其权限,只需对用户进行角色分配,用户的权限由其角色所规定;其次可以合理划分职责,不同的角色所负担的职责不同,其权限也就不同,这样可以保证用户都是进行自身职责范围内的操作;再次可以防止权力的滥用,例如有的操作需要多种角色的用户合作才能进行,单个用户不能顺利执行。

一种可行的用户角色划分方法是将用户划分为领导者、合作者、参观者、书记员这四种角色。领导者:一个任务的负责人,具有最高的权限。例如领导者可以决定将发言权交给哪个用户,可以决定其他用户角色的分配;合作者:主动地参与完成某一任务的一般用户,与同组中的其他用户进行合作;参观者:不承担具体的工作而只是观察工作进展,提出意见。例如可以观看白板,但不能在白板上进行操作;书记员:不参与任务的完成,但是负责记录工作进程和保存工作结果。

由于协同系统中,用户经常会发生变动,因此系统还必须具有动态配置的功能。对于中途退出的用户,如果存在权力移交的问题,应该妥善处理。例如领导者退出的情况下,要重新指定一个领导者。

并发控制:当有多个用户同时发生的动作产生冲突时,要有合适的并发控制机制来解决。例如用户对白板上的某个图做说明,欲画一个箭头,当该用户正在画该箭头时,可能另外一个用户将白板翻至第二页,这样就会出现不一致的情况^[6]。图2显示了可能出现的情况。(a)用户本来的意图;(b)当出现冲突时,箭头被画在第二页上;(c1)、(c2)为另一种可能的冲突情况,箭头的一部分被显示在第一页,另一部分被显示在第二页。

现有的并发控制方法有很多种,其中有一些并

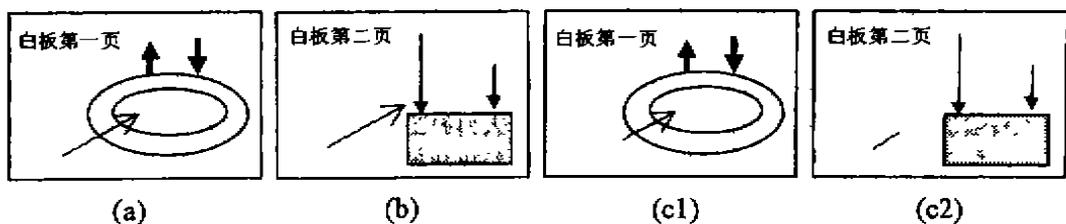


图2 用户操作发生冲突的情况

发控制方法是基于冲突检测的,例如 GROVE^[1]中的操作转换;还有一些是冲突避免的方法,例如

CES^[7]中的读/写锁和 Tickle 锁,CoNus^[3]中的 Floor 控制。基于冲突检测的方法是一种乐观的方

法,在用户的操作执行之后再检测,如果与别的用户的操作发生了冲突,则进行回退或者操作转换,这种方法是建立在冲突发生的可能性不大的乐观假设的基础之上的,基于冲突避免的方法是一种悲观的方法,在用户的操作执行之前,检验是否会发生冲突,这种方法是建立在冲突发生的可能性较大的悲观假设的基础之上的。

在白板上,可以采用基于冲突避免的集中式锁机制来解决如图2所示的冲突问题,在用户可能会引起冲突的操作(例如翻页)执行之前,检查操作对象(白板)是否加锁,如果未加锁则锁住该对象,其他用户就不能使用,操作执行结束之后再对其解锁。在协同编辑器中,也可以采用这种锁机制,可行的一种实现方法如下:以自然段为锁的单元,保存一张段落表用于记录被锁住的段落。当用户欲对某一段进行编辑时,发出申请,根据申请编辑的段落序号,查找段落表,如果被锁住则拒绝该申请并提示用户稍后重试,否则授权该用户进行编辑,并且将该段落登记入段落表中。在操作过程中如有改变段落的事件发生,如增加一段,删除一段等,锁住的段落序号也相应要进行改变,这需要修改段落表。在语音和视频工具中,并发控制机制相对简单,每一时刻只能有一个用户在发言,且系统转发该用户的视频图像至其他协同用户,由担任领导者角色的用户来决定发言权的分配和收回。领导者还可以选择采用时间戳方法让每一个用户轮流获得一个时间片,在此时间片内该用户拥有发言权。

用户中途加入的处理(latecomer):对于一次已经在进行中的会话,中途加入的用户应该获得当前系统状态信息例如白板的当前内容,以便加入到合作中来,有两种方法可以实现当前状态的获取。一是采用事件日志,记录下所有用户的操作事件,当新用户中途加入时,将此日志传递到新加入的节点,进行事件回放;二是将整个系统的当前状态如白板的像素或编辑器的文本传递到新节点^[6]。前一种方法需要缓冲区但是传输的信息量少;后一种方法不需要缓冲区但传输信息量大。新用户中途加入时,获取当前状态的时候,其他用户仍然在进行操作,这就要处理好在状态获取过程中发生的事件,否则会造成信息的丢失,解决这个问题的一种方法是在新用户加入时不允许其他用户进行操作,直到该用户成功加入进来;另一种方法是采用事件日志,记录下从用户开始获取信息到成功加入这一段时期内所有用户的操作事件,待该用户成功加入之后将日志传递过去

进行回放。两种方法各有优缺点,前一种方法如果用户加入过程耗时较多,就会造成会话的中断,后一种方法要为每个中途加入者开辟额外的缓冲区。

多媒体信息的同步:同步也是桌面文档会议系统中需要解决的重要问题,由于参与会议的各结点其机器处理能力和网络带宽、时延等特性的差异,造成了各个用户的应用程序状态的不同。如果没有适当的同步机制,会造成不一致。例如,当一个用户在白板上打开一幅设计图,并在其上进行修改时,此时可能有某个用户还没有接收并显示完这幅图,这个用户就只能见到前者在白板上所绘制的点线,不能理解前者的操作意图,也就不能有效地参加到会议中去。因此同步对于信息的感知和状态的一致是至关重要的,实现同步的方法有不少。例如在上面的例子中,可以实现一种同步机制使得只有当所有的用户都接收并显示完这幅图才允许进行编辑。这种方法对于各结点的网络特性和处理能力都接近的情况比较合适,但如果有一部分用户的网络速度要比其他用户慢很多,就会大大降低整个系统的效率,这时应该考虑建立一个缓冲区,记录下用户的操作,当速度较慢的结点接收完图像时,再从缓冲区中读取操作记录予以回放。同样地,对于视频和语音信息也要进行同步。例如,有几个用户轮流发言,当一个用户发言时,其他用户通过视频窗口看到该用户正在说话,并且也能听到他说的话,但如果视频和语音信息的同步没有处理好,有可能某个用户的视频窗口显示的是某个用户正在说话,而语音信息却还没有接收完毕,等到语音接收完毕听到该用户说的话时,视频窗口却已经在显示下一位发言者了。

结束语 本文针对桌面文档会议系统,从体系结构、访问控制、协同感知、角色管理、并发控制、中途加入、同步等几个方面,详细分析了其实现中的关键技术,桌面会议系统可以广泛应用于诸如网络会议、远程教学、虚拟实验室、远程医疗系统当中,使得高复杂度的协同工作成为可能。CSCW的研究已经成为当前的热点,实用化的协同系统将使计算机在人们的工作中发挥越来越重要的作用,目前我们已经成功地开发了一个实用型的CSCW系统——计算机支持协同科研系统CSCSR,该系统集成了前述桌面文档会议系统的多项功能,并以科研合作为背景,开发了虚拟设备操作的功能,创造了虚拟实验室的工作环境,该系统提供给南京大学固体微结构国家重点实验室与东南大学分子电子学开放实验室的合作科研项目使用。

(下转第73页)

会影响用户的阅读效果。

3.4 信息安全

Internet 是无政府的社会,每个人都可以在上面享受到高度的自由,这就使得网上信息良莠不齐,信息垃圾充斥于网上并可不受限制地到处传播。

广存系统是通过多级接力式的 push agent 发布信息的,这就使得信息的发布可以得到控制,过滤掉信息垃圾,每一级 push agent 都设有一个可信度,用以供下一级 push agent 进行信息过滤。当某 push agent 的可信度高时,由它推送的信息可直接向前推送;反之,则需对信息进行按内容检查,符合要求的才能向前传送。

3.5 基于内容的超媒体信息管理

超媒体使得对同样的内容可以用文本、图像、声音、视频等多种媒体形式表达,它们都属于复杂数据,因此把它们以内容为中心进行链接、存贮、查询及其它管理是一种有效的方法。信源将要播送的内容以一种或几种媒体形式表达出来,并按内容建立基本链。客户机按用户的需求从 CATV 信道中自动抓取数据,进行裁剪,存入自己的 GB 硬盘中。硬盘中存贮的媒体是物理媒体,相应地,在用户查询时按内容生成的临时的媒体形式为逻辑媒体。这样用户查询时,可在基本链的基础上按内容生成动态链,并

可以根据用户的要求用不同的逻辑媒体形式作为查询结果。这样,就使以内容为中心的超媒体信息管理具有极大的灵活性、适应性和动态性。

结束语 本文提出了一种新的信息共享方式——PUSH 技术支持的分布信息共享机制。该模型有效地利用了计算机网 IP/PUSH 技术、电视网广播信道和高带宽的优点。在用户提出查询要求前,数据已被预读入 GB 硬盘中,使用户从提出要求到得到结果的响应时间大大缩短。目前,我们已经突破了 10×10^7 bps 在普通 CATV 信道中传输及家庭数据库软件等关键技术,通过了 60 户网软、硬件的实际测试,下一步将进行 300 户网及更高速率数据传输系统的研究与实验,经一年来的试验运行证明,它不失为一种对传统 Internet(计算机 IP 技术+公共电信网)设施的良好补充。

参考文献

- 1 陈彦,徐宏炳,王能斌. WWW 分布数据源研究——数据模型和查询语言. 软件学报, 1998
- 2 李光亚,周学海,等. 一种基于语义网络的开放式超媒体系统结构. 计算机科学, 1997
- 3 李幼平. 用“广存”机制帮助 INTERNET. 计算机世界, 1997

(上接第 66 页)

参考文献

- 1 Ellis C A, et al. Groupware: Some Issues And Experiences. Comm. ACM, 1991, 34(1): 39~58
- 2 Palmer J D, Fields N A. Computer-Supported Cooperative Work. IEEE Computer, 1994, 27(5): 15~17
- 3 Bentley R, et al. Architectural support for cooperative multiuser interfaces. IEEE Computer, 1994, 27(5): 37~45
- 4 Zhao J, Hoppe H U. Supporting flexible communication in heterogeneous multi-user environments. In: Proc. of the Intl. Conf. on Distributed Computing Systems. 1994
- 5 Reinhard W, et al. CSCW Tools: Concepts and Architectures. IEEE Computer, 1994, 27(5): 28~36
- 6 Shirmohammadi S, et al. Applet-Based Telecollaboration: A Network-Centric Approach. IEEE Multimedia, 1998(April-June): 64~73
- 7 Greif I, et al. A Case Study of CES: A Distributed Collaborative Editing System Implemented In Argus. IEEE trans. on software engineering, 1992, 18(9): 827~839
- 8 Mao B, Xie L. An Object-based Model for Prototyping User Interfaces of Cooperative System. ACM Software Engineering Notes, 1997, 22(2): 72~76
- 9 茅兵, 谢立. 基于对象的协同计算模型. 中国科学(E 辑), 1997, 27(6): 542~547