

CSCW 系统中的应用共享技术研究

Research on Application Share in CSCW System

许玲 (广东技术师范学院电子信息工程系 510631)

摘要: 本文分析了 CSCW 系统中应用程序的两种主要的共享技术,“共享显示法”与“事件分发法”,提出了以“事件分发法”作为应用程序共享方式的一个系统模型实例,详细分析了该模型的结构与通讯方式,并提出了一种利用客户端插件(Plug-in)来协调客户、客户和服务以及客户和客户之间的通信机制。

关键词: 应用共享 CSCW

1 概述

一个 CSCW 系统的成败很大程度上取决于它的应用程序的共享情况,所谓应用共享技术是指由一个群体的各成员通过各自的机器共同控制在一台机器上执行的应用程序,使之可以由多个用户共同控制、实现协作。采取何种策略、如何将单用户应用程序做尽量少的修改或者不加修改地在 CSCW 环境下运行,且提供较完善的协作服务是应用程序共享技术的目的。扩展已有的大量单用户应用程序,使之变成 CSCW 体系下的可实现多用户协作的应用程序,这也是为了保护开发单用户应用程序所进行的大量已有的投资,因此,应用程序的共享技术是现在各种 CSCW 系统研究的重点内容之一。

2 应用共享技术分类

按照时间特性对 CSCW 系统进行分类[4],协同工作既可以是同步的,也可以是异步的,同步方式的协同操作表示合作者之间是紧耦合关系,即某个合作者的操作立即可以被群组中其他成员感知。而异步方式表示合作者之间是松耦合关系,合作者的操作结果不是立即送给其他成员,而是在一段时间后才被其它成员知道。关于异步协作系统,在此不作深入讨论,因为异步协作系统不涉及到应用共享的问题,在这里主要讨论同步协作系统,从应用程序的协作方式上对同步协作系统进行分类,可分为两大类,“共享显示法”与“事件分发法”。

“共享显示法”的体系结构是这样的,只有一个机器上运行着应

用程序,会议代理机构复制用户的输入与应用程序的显示,然后把它复制到参加会议的所有用户那儿,并按一定的策略合并各用户的输入,以达到共同控制应用程序的目的;这种方法与具体的应用程序无关,单用户应用程序可以不加修改地在这种系统下运行。图1是共享显示法结构示意图,这种系统包括Microsoft Netmeeting、Intel Proshare 及 Sun ShowMe等。从功能的角度来看,这种系统实现的是“透明协作”,即WYSIWIS(What You See Is What I See)。这种模型结构是非常简单的,没有弹性的会议控制,没有对安全机制的支持,没有对任何形式的异步协作的支持,协作过程也不能重放等。

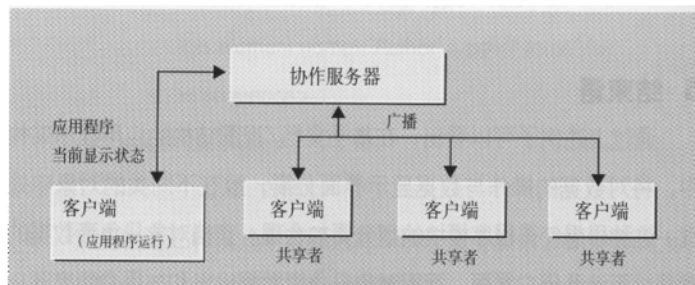


图1 “共享显示法”结构示意图

共享显示系统对带宽的要求非常高,因为全部的显示信息要传到所有的参加会议者那里,且传输错误容易导致参加者的显示状态不一

致, 总之, 对于连接速度较慢的情况下, 这种形式显然是不可行的。

“事件分发法”的系统只在网络中交换应用程序状态改变的事件, 每一个参加协作的机器都运行一个本地的应用程序的副本, 在网络中只传送使应用程序状态改变的事件, 而不是整个应用程序的显示, 该方法克服了共享显示方法的缺点, 但是由于事件分发法涉及到应用程序的内部, 开发者需要决定什么事件需共享, 什么时间以及如何共享等问题, 显著增加了开发的难度; 这种体系结构可进一步细分为两实现机制, 分别是“全部事件共享”与“部分事件共享”, 其中“全部事件共享”实现的也是“透明协作”, 在这种体系结构中, 所有的事件是共享的, 该方式也可以让单用户应用程序不加修改地由一组人使用, 该结构是通过修改Windows toolkit来实现的, 以便事件被送到本地机之前被截获, 然后送到协作者那里, 事件分发法中的“全部事件共享”机制与“共享显示法”实现的都是“透明协作”, 但它们实现的原理是不同的。在“部分事件共享”结构中, 开发者可以决定哪些事件可以共享? 什么时候共享? 如何共享? 因此对这类系统可以进行弹性控制、支持安全机制、支持异步协作、协作过程可以记录可以重放等。

事件共享系统的一项重要能力是可以记录系统的活动, 因为所有的系统与应用程序的消息通讯都需要通过中心协作服务器, 每当控制消息或者是应用程序消息经过中心协作服务器时, 它都被记录到一个数据库中, 同时该消息发生的日期、时间, 还有发送者的信息都记录下来, 这些数据可以事后访问它, 也就是说, 整个系统的活动可以异步地再现。在系统重放的过程中, 数据库就作为一个信息源, 因为它里面已保存了应用程序消息与控制消息, 用户的状态、会议的配置与应用程序的运行都可以被恢复, 这是因为储存了应用程序的消息也就是保存了应用程序里面的活动。

事件共享系统在过去被认为是不可能实现的, Web结构使得事件共享应用成为可能, 主要有以下重要的因素保证了事件共享的实现: HTTP与Java, Javascript 的下载机制保证了软件模块的实时分发; Java虚拟机为所有的应用程序保证了统一的独立平台环境; 面向对象的程序设计方法为地处分布的应用程序实时共享事件状态提供了保障。

3 “事件分发法”协作系统的一种模型框架

“事件分发法”的协作系统的目的是将基于Web的应用程序与非基于Web的应用程序低成本地集成为CSCW应用系统, 核心问题之一是应用程序产生的“事件”如何截获并且传送到协作者的主机中, 并且该“事件”能在协作者的本地应用程序副本中产生相应的操作结果, 一般来说, 处于共享中的应用程序每当有按钮被击时就产生了一系列需共享的“事件”。核心问题之二是这一系列的“事件”能否存储并且重放, 这项功能一方面可以解决“迟到者”的问题, 另一方面

可以事后访问它, 也就是说, 整个系统的活动可以异步地再现。

3.1 模型框架

本文提出了一种基于“事件分发法”协作系统模型框架如图2。为了能使基于Web的应用程序(例如以Java Applets形式存在的应用程序实例)或其他形式的应用程序实例之间通信, 模型扩展了Web浏览器的功能, 与浏览器插件及服务器端软件共同组成了一个交互式的多媒体协作环境; 在这个环境下可以构建实现各种应用功能的协作系统。

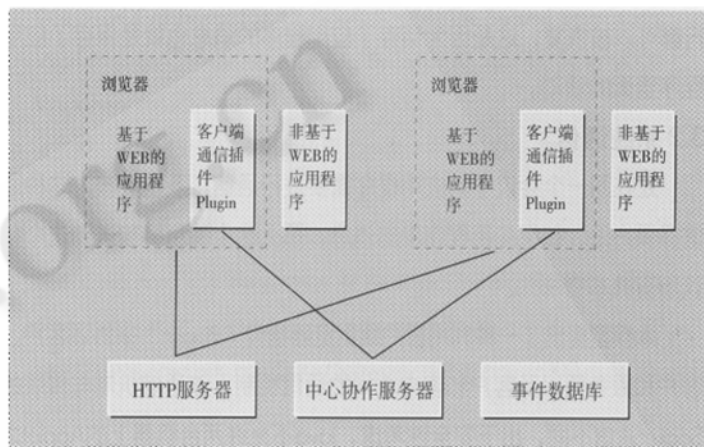


图2 协作系统模型框架

客户端通信插件(Plug-in)的主要任务是中转应用程序与中心协作服务器的通信, 启动本地应用程序, 它是独立于操作系统的部分。中心协作服务器是主要的通信部件, 是服务器端软件, 所有客户端通信插件(Plug-in)都与中心协作服务器通信, 中心协作服务器维护系统状态数据(参加者、应用程序和会议等), 它的主要任务是维护整个系统的状态, 为参与每一次会议的应用程序之间的消息传递寻找路径, 并将事件/数据记录到后端数据库中。中心协作服务器监听所有无会议管理者与其连接, 每当有一个连接时, 它就启动一个线程, 它把所有与其连接的会议管理者的全部信息都存储在动态数据结构中, 同时也将这些信息的第一步变化都存入后端数据库中(以备实现异步协作)。

非基于WEB的应用程序指所有独立运行的用户程序。可用任何编程语言编写, 通过API与客户端通信插件(Plug-in)通信。Plug-in负责非基于WEB的应用程序的启动和对外通信, 基于WEB的应用程序指用Java编写的用户程序Java applets, 从HTTP服务器上下载, 在浏览器中执行。Java applets通过调用客户端通信插件(Plug-in)的方法彼此进行通信。

系统支持事件的分发, 通过插入API调用, 开发者能够指定要分发的的事件, 如何分发和分发到哪里去则由会议管理层决定。

对下述用户动作, 控制程序将产生一系列的专门事件, 如进入系统、退出系统、激活运行本地程序、离开会议、加入与新的应用程序的会议、加入与已有的应用程序的会议、激活运行远程应用程序、切

换至Master模式（或者说发言权控制事件）等等。

在该模型框架下可以记录系统的活动，因为所有的系统与应用程序的消息通讯都需要通过中心协作服务器，每当控制消息或者是应用程序消息经过中心协作服务器时，它都被记录到一个数据库中，同时该消息发生的日期、时间，还有发送者的信息都记录下来，这些数据可以事后访问它，也就是说，整个系统的活动可以异步地再现。在系统重放的过程中，数据库就作为一个信息源，因为它里面已保存了应用程序消息与控制消息，用户的状态、会议的配置与应用程序的运行都可以被恢复，这是因为储存了应用程序的消息也就是保存了应用程序里面的活动。

3.2 系统通信

处在同一个会议中的应用程序需要进行通信，系统应该为这种通信提供方法，如果应用程序发出消息，那么这个消息将被送到这个会议所有共享者那儿。

本模型提出了一种利用客户端浏览器插件（Plugin）来协调客户、客户和服务以及客户和客户之间的通信机制，该通信机制与传统的Socket通信不同，出于安全性考虑，Java不允许两台机器上的Applet之间直接通信，由于Java Applets的这些安全性限制，要实现Java Applets应用程序实例之间的通信，就必须有第三方来介入与支持，在该系统中设计了一个位于客户端的通信中间件Plugin，它负责中转Java Applets与协作服务器之间的通信。

在系统的工作过程中主要有两类消息，控制消息与应用程序消息，控制消息主要发生在客户端通信插件(Plug-in)与中心协作服务器之间，这类信息主要的功能是为一个用户登记参加会议、创建一个会议、启动应用程序等等，控制信息对用户而言是不可见的。应用程序消息主要是在用户的应用程序之间发送，应用程序信息的结构依赖于应用程序本身，通信系统不对它进行分解，对应用程序消息而言，通信系统是透明的，应用程序消息的组织是利用API去定义的，图3是通信示意图。

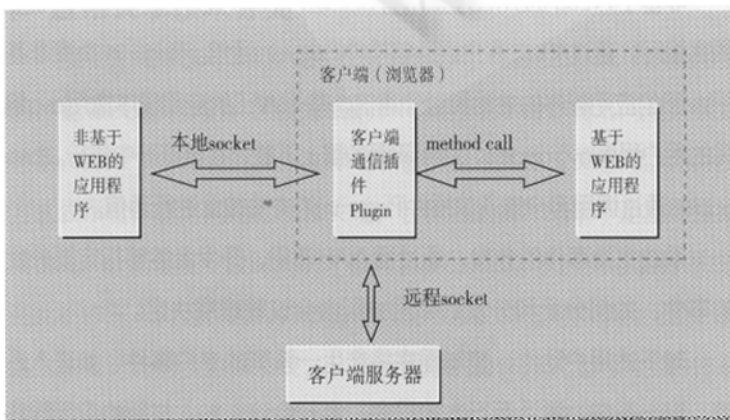


图3 通信示意图

3.3 控制权转换机制

应用共享中所共享的一股是第三方单用户可执行程序，共享用户很难利用对象加锁和区域加锁等并发控制机制，因此应用共享中为保证共享对象的一致性，并发控制策略一般是控制权的转换机制，即Floor Passing机制[3]。这种策略是基于“全部/什么都没有”（all-or-nothing）的思想，只有拥有控制权的成员才有权利操纵共享的应用。同一时刻只允许一个成员进行操作，其他成员观看。一般的控制策略如下：(1) FIFO的策略：设置一个申请队列，按照申请的先后顺序在队列中排序，每次在申请队列头的成员可以获得控制权；(2) Master授权模式：由Master指定某成员获得控制权，该成员操作完成后将控制权归还给Master模式；拥有控制权的成员可以将控制权直接传给另一个申请者；(3)协商环方式：控制权按照事先定义好的路线自动传递；(4)抢夺式：控制权释放后，成员进行抢夺。

4 其他的模型与系统

JCT[Java Collaborator Toolset][2][3]是美国Old Dominion大学开发的一个Java应用程序的协作平台。通过一个被称为Collawt的新的AWT类库替代Java原有的AWT类库，JCT能把一个单用户界面的应用程序变成一个多用户界面的应用程序，从而支持“透明协作”（Transparent Collaboration）。Collawt具有在协作者之间自动分发事件的功能，即Collawt解释所有的用户事件，并将其送到事件分发器，由事件分发器向协作组内该应用程序的所有拷贝广播。该系统也采用“事件分发法”作为它的应用程序共享技术。

Habanero[5]来自著名的NCSA，是基于Java的面向对象分布处理系统，由于系统全部用Java application编写，从功能上看，既能支持并行计算机，又能实现分布处理。系统扩充了Java对象库，解决了多线程存取共享存储变量的同步问题，提供了基于消息传递和基于分布式共享存储等多种编程接口，实现多种计算模式，该系统也采用“事件分发法”作为它的应用程序共享技术。

多媒体会议系统属于CSCW的一种应用形式，Microsoft的Netmeeting可以称为此类软件的代表，应用程序共享是Netmeeting的一项重要功能。可以与其他与会者共享运行在某一台计算机上的程序。与会者可以再次浏览相同的数据或信息，而且观察用户在共享应用程序时对该程序所执行的操作（例如，编辑内容或翻阅有关信息）。与会者还可以透明地共享基于Windows的应用程序，共享应用程序的人还可以选择与会议中的其他人协作，他们可以轮流编辑或控制应用程序。Netmeeting是采用“共享显示法”实现应用程序协作的，这种方法的特点是只有一台机上运行有应用程序，由于受Netmeeting所采用的共享机制的限制，Netmeeting的使用者在同一时间只能参加一个会议，且在会议过程中不支持异步方式的协作，即不支持会议记录与重放。

5 结束语

本文分析了CSCW系统中应用程序的两种主要的共享技术，“共享显示法”与“事件分发法”，提出了以“事件分发法”作为应用程序共享方式的一个系统模型实例，详细分析了该模型的结构与通信方式，并提出了一种利用客户端插件（Plug-in）来协调客户、客户和服务器以及客户和客户之间的通信机制。

应用共享技术是一个CSCW系统成败的关键，现在的两种主要技术“共享显示法”与“事件分发法”各有优缺点，“事件分发法”所支持的系统虽然可以进行弹性控制、支持安全机制、支持异步协作与协作过程的重放等功能，克服了共享显示方法的缺点，但是由于事件分发法涉及到应用程序的内部，开发者需要决定什么事件需共享，什么时间以及如何共享等问题，因此，需要在应用程序内部增加一些协作代码，显著增加了开发的难度，对于一些无法得到其源代码的应用程序也不能采用“事件分发法”的方式进行共享。

参考文献

- 1 NPAC, Syracuse University Tango project Web site
<http://trurl.npac.syr.edu/tango>, 2000。
- 2 S. Valin, A. Francu, H. Trefftz, and I. Marsic, Sharing Viewpoints in Collaborative Virtual Environments,[C] In Proceedings of the 34th Hawai'i International Conference on System Sciences (HICSS-34), Maui, Hawai'i, January 3-6, 2001。
- 3 M. Ionescu and I. Marsic, An Arbitration Scheme for Concurrency Control in Distributed Groupware,[C] In Proceedings of The Second International Workshop on Collaborative Editing Systems, An ACM CSCW'2000 Workshop, Philadelphia, PA, December 2000。
- 4 史美林、向勇，《计算机支持的协同工作理论与应用》[M]，电子工业出版社，2000。
Shi Meilin, Xiang Yong 《Computer Supported Cooperative Work theory and application》, Publishing House of Electronics Industry, BEIJING 2000
- 5 Mary Pietrowicz , Synchronous/Asynchronous Blending in Habanero [R]
<http://www.ncsa.uiuc.edu/SDG/Software/Habanero>, 2000。