

# 基于 MVC 的协作多用户交互接口设计与实现<sup>\*</sup>

李军怀<sup>1</sup> 张彤<sup>1</sup> 彭军<sup>2</sup> 张景<sup>1</sup>

(西安理工大学计算机科学与工程学院 西安 710048)<sup>1</sup>

(重庆科技学院电子信息工程学院 重庆 400050)<sup>2</sup>

**摘要** 针对目前 CSCW 环境中用户端设备的多样性、用户个性化需求特点,通过对协作多用户接口特性、支撑结构和实现方法的分析,提出了一个便于裁剪、面向用户的 MVC 结构的协作多用户交互接口模型。在此模型中,充分利用 MVC 结构视图与控制分离的特点,将协同工作环境中共享信息集中存放于共享空间中,维持一致性;而信息显示和人机交互的语义部分复制在分布的用户接口视图中,使得参与协同工作的多个用户可以采用不同的设备或裁剪自己的显示界面。

**关键词** CSCW, MVC, 多用户接口, 代理

## Design and Implementation of Cooperative Multi-user Interface Based on MVC Design Pattern

LI Jun-Huai<sup>1</sup> ZHANG Tong<sup>1</sup> PENG Jun<sup>2</sup> ZHANG Jing<sup>1</sup>

(School of Computer Science and Engineering, Xi'an University of Technology, Xian 710048)<sup>1</sup>

(School of Electronic Information Engineering, Chongqing University of Science and Technology, Chongqing 400050)<sup>2</sup>

**Abstract** This paper presents an easy to tailor, user-oriented cooperative multi-user interface model based on the analysis of cooperative multiuser interface characteristics, supporting structure and implementation method, which is adaptable to the diversity of user devices, user customization. This model makes full use of the separation feature of view and control in MVC design pattern and stores the share information to share space centrally to maintain data consistency. The semantics of information display and human-computer interaction is replicated in distributed user interface Agent such that allows end users of an application easy to tailor their user interface and use different devices.

**Keywords** CSCW, MVC, Multi-user interface, Agent

## 1 引言

在任何一种计算机应用系统中,接口技术作为行为动作的确切理解和表达手段起着十分重要的作用。特别在计算机支持的协同工作(CSCW)环境中,为了支持多层次、多群体的协同工作,人机交互接口除了具有人机交互的功能外,还要反映并协调协作群体的活动。与单用户系统相比,协作多用户接口提供了一种更加高效的工作方式,系统必须允许多个用户同时或先后访问应用系统,提供方便的多用户接口,使得用户请求与所需信息可以通过用户接口表示并传递到系统中,而请求结果也可以通过用户接口反映给用户。

目前,对协作多用户接口还没有一个严格的定义,因为它只是 CSCW 系统中的用户接口部分,不同的应用,其用户接口有不同的功能和表现定义,即协作多用户接口的明确定义依赖于具体的 CSCW 系统的软件体系结构模型。但总的来说,协作多用户接口的目标是建立和维护一个公共环境(Common context),以允许一个用户的活动能够反映在其它用户的屏幕上。这个公共环境是通过共享应用信息获得的,实时表现和对共享信息的操作是协作多用户接口的主要功能<sup>[1]</sup>。

基于特定的应用,人们对多用户接口进行了许多有益的

探讨,但由于多用户接口的复杂性,目前还没有能够适合各种环境和各种应用的通用的多用户接口<sup>[2~4]</sup>。特别是,随着无线网络和许多便携式计算设备的广泛使用,要求多用户接口应用系统能够适应异构计算环境以及用户个性化要求。本文从参与协作的用户端设备的多样性以及用户个性化需求出发,通过对 CSCW 环境中的用户接口特性、支撑结构和实现方法的分析,提出了一个便于裁剪、面向用户的 MVC 结构的协作多用户交互接口模型,并应用到了实际系统之中。

## 2 协作多用户接口分析

### 2.1 特性分析

在传统的人机接口模型中,用户接口只接受本地用户动作的驱动,并且只反映出动作的回显和计算结果的显示,因而只注重对用户动作的理解,而对于协同的用户接口,某一用户动作可以驱动多个用户接口,同时一个用户接口可以反映多个用户动作,用户动作结果需要多种表达方式,表达与理解是协作系统用户接口要实现的一对互逆过程,并且一个用户对某一用户动作理解到一定层次,需交由另一用户接口去以其相应的要求表达。

协作多用户接口形式如图 1 所示,其具有如下特性<sup>[2,5]</sup>: 分布性、人-人交互特性、协作与自治性、多媒体表现特性、支

<sup>\*</sup> 基金项目:国家 863 资助项目(编号:2002AA414060)和陕西省教育厅产业化培育资助项目(编号:01ZC24)。李军怀 副教授,博士,主要研究方向为分布式计算、CSCW;张彤 讲师,主要研究方向为 Internet 应用;彭军 副教授,博士,主要研究方向为分布式计算及信息安全;张景 教授,博导,主要研究方向为分布式计算及 Internet 应用。

持不同要求的共享操作过程和开放性。

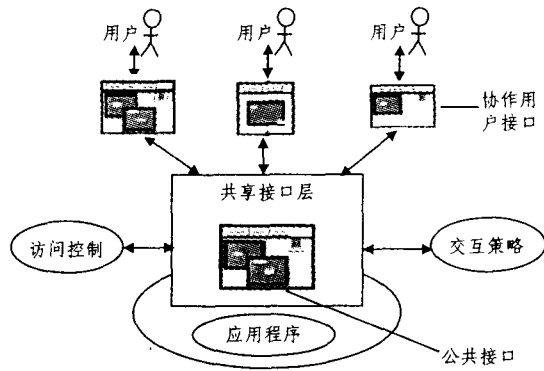


图1 多用户接口形式

## 2.2 现有多用户接口实现方法

目前,在协作多用户接口的设计与实现中,有三种方法<sup>[6]</sup>:

(1) WIMP 方法。标准的 WIMP (Windows, Icons, Menus, Pointer) 方法是当前 GUI 的事实上的标准,当前许多群接口也是基于 WIMP 方法提供多重窗口。

(2) Web-based 方法。将 CSCW 与 WWW 的集成,一方面在现有的 Web 接口中增加协同工作能力,另一方面研究在现有的 CSCW 系统中引入 Web 技术。

(3) VR 方法。利用 VR 技术为 CSCW 系统提供一个全新的、虚拟的多用户交互环境。

在协作系统接口的实现中,大多数系统采用的是 WIMP 方法。这是因为对于 Web-based 方法,虽然由浏览器定义的接口能够跨平台、统一、简单而目直观。但其缺点是限制了用户接口的灵活性,对接口设计带来一定的限制(如直接操作、即时反馈和施放操作等高级接口用 HTML 是无法实现的),协作感知的即时性实现困难等。VR 方法虽然是一个非常出色的辅助手段,但是目前从硬件和软件等多面还存在一定的困难。

## 2.3 传统的支撑结构

CSCW 环境中用户接口所追求的目标不同于分布式系统中,用户仅仅考虑如何用计算机去支持他自己的那部分工作,它关注的重点更多地在于如何用连成网络的计算机系统去支持多个人构成的群组进行随时随地的交流与协作。目前,采用的接口模型有集中式、复制式和混合结构三种<sup>[2,3,7]</sup>。

(1) 集中式结构 (Centralize Architecture) 这是一种共享屏幕或共享窗口的接口方式,多个用户接口只与一个应用交互,其间的关系是任意的、无结构的连接。用户之间的动作协同是通过输出的广播不自觉地实现的。没有专门的协同处理,因而这类用户接口也称为协同透明 (collaboration transparency) 的用户接口。优点是:简单和易于实现;简化了事件的调度;可以方便地实现很细粒度的协同。缺点是:缺乏灵活性;健壮性差;响应时间长。适用于交互性不是很强的 CSCW 环境中,如各种异步系统。而各种支持同步协作的系统很少采用这种纯粹的集中式结构,特别是在广域网络环境下。

(2) 复制式结构 (Replicated Architecture) 复制式将应用程序复制在所有的参与工作的用户节点上,在每一个用户的工作站上都有共享应用的一个副本。为了保证应用的不同副本之间的同步,所有输入应传送给应用的每一个副本,并且应有一个并发控制机制,以保证应用的不同副本以相同的序

列处理用户输入事件。使用这种结构的例子有 Groves 和 CES<sup>[8]</sup>。相对于集中式结构,其优点是可靠性较高,响应速度有所提高。缺点是系统复杂性提高,开发困难,需要考虑同步控制等问题。适应于一些交互性比较强的场合,如各种实时的同步协作系统。

(3) 混合结构 (Hybrid Architecture) 由于集中式结构难以支持不同级别的信息共享,共享操作的实时性难以保证,而复制式结构又难以实现信息的一致性和成员的动态加入与撤离,因此这两种结构很难满足通用 CSCW 用户协作接口的需求。

混合式介于集中式和复制式之间,所有用户只有一个应用,但每个用户都有一个对话模块。它将接口的对话模块复制,用以处理一般的交互动作,对话模块与应用之间的接口是对象,所以也称为共享对象的用户接口,它作为前两种的权衡,得到了广泛的使用,但与前两种相同的是,多个用户接口之间的连接只限于某一层,即只有一条信息通道,混合式中用户交互动作只经过对话模块处理,提高了反馈速度,但无法传达给其他用户,降低了协同的效率。典型的实验系统有多用户 Suite 和 Colab<sup>[8]</sup>。

## 2.4 存在的问题

从协作过程的时间限制特征来分,协作系统分为同步协作和异步协作两种形式。一般而言,实时群件系统大都采用了一种被称为是“你所见即我所见”(What You See Is What I See, 即 WYSIWIS) 风格的多媒体用户界面,例如协作诊断会议中的白板、医学图像协同标注工具等都是这种风格界面的典型例子。它实际上给各协作用户提供的是关于共享环境中对象的相同的视图。这种严格的 WYSIWIS 风格的界面在各协作参与者基本对等(即具有基本相同的知识背景、处于基本相同的协作地位、扮演基本相同的角色等)的情况下基本上是合适的。目前的 CSCW 协作多用户接口基本采用此方式,然而这种方式无法满足用户端设备多样性以及个性化需求,并可能会引起以下问题:

(1) 可能会引起一些操作上的不便,例如当某个用户正在处理共享对象的某一部分时,其他用户对视图的滚动可能会导致这部分被处理的内容移出窗口范围外;

(2) 由于有多个用户同时对共享对象进行处理,因此用户的视图可能会发生频繁的动态变化,例如当一个用户正在文档中某处输入字符时,其他用户删除其插入点前面的字符或在前面插入新的字符均会引起前一用户的插入点位置的改变;当三个用户同时移动白板中的三个不同的因素时,屏幕上可能会有三个不同的对象同时发生移动,这些都可能会引起屏幕内容的杂乱无章并分散用户的注意力;

(3) 当协作用户具有不同的背景显示设备,并且关注的内容也都各不相同,这种单一的视图很难满足这些用户的不同需要。

为了解决这些问题,目前已经有许多的文献<sup>[2~4,7,9]</sup>从多个不同的侧面对 WYSIWIS 进行改造,提出了解决方案。例如可以对视图中对象的显示位置、同步的时间以及同步的方式进行调整,或者对视图的种类进行丰富、对视图中显示的内容进行筛选等等。而在异步 CSCW 系统中,一般使用的是一种类似于电子邮件系统的用户界面。这种形式的界面最大的好处就是简单,用户只需象流水线上的工人那样机械地处理“流过”他面前的任务即可。但其共同的特点是在整个协作过程中,用户只是“被动”的信息接受者与处理者。系统既没有

或者很少给他们提供协作过程的整体状态信息,也没有提供一种有效的途径让用户知道他在协作过程中的整体位置与作用。在实际的工作过程中,用户可能会根据他自己的职责、与周围的人的关系等现实的工作环境而采取比较灵活的处理手段。同时随着计算机技术的发展和用户需求的变化,多用户接口还应该满足多用户显示的支持、不同视图的支持以及用户裁减性的支持等需求。

### 3. 基于 MVC 的多用户接口模型

根据上述对多用户接口模型的分析,从目前用户界面设备、显示方式的异构性和多样性出发,在此,本文借鉴 Model-View-Controller(MVC)设计模式,提出一种介于集中式结构和复制式结构之间的一种混杂结构:共享信息集中存放于共享对象集合中,维持一致性;而信息显示和人机交互的语义部分复制在分布的用户显示代理(Agent)中。用户显示 Agent 封装了用户接口管理的共享策略,实现协作感知的信息共享<sup>[5,10,11]</sup>。

#### 3.1 传统的 MVC 结构

传统的面向对象编程中,MVC 设计模式将一个交互应用分成三个部分:

(1)View:用户视图模块,这是用户界面部分。管理应用程序与用户之间的接口:一方面,它为用户提供了输入手段,并触发应用逻辑运行;另一方面,它又将逻辑运行的结果以某种形式显示给用户。

(2)Controller: 流程控制模块,该部分是用户界面与 Model 的接口。一方面它解释来自于 View 的输入,将其解释成系统能够理解的对象,同时它也识别用户动作,并将其解释为 Model 特定方法的调用;另一方面,它也处理来自于 Model 的事件和 Model 逻辑执行的结果,调用适当的 View 为用户提供反馈。

(3)Model:事务逻辑模块,这是整个模型的核心。它表示的是解决方案空间的真正逻辑。它采用面向对象的方法,将问题领域中的对象抽象为应用程序对象。在这些抽象中封装了对象的属性和这些对象所隐含的逻辑。

可以看到,MVC 编程模型把用户界面设计、流程控制和事件逻辑进行了分离。

#### 3.2 基于 MVC 的多用户接口模型

基于上述的分析,本文提出如图 2 的用户接口模型。

用户接口 Agent 管理所有用户接口的共享信息,在这些共享信息上定义一些视图,用户通过视图浏览和修改共享信息,每个用户屏幕是视图集的一个子集,即工作集(Working Set)。每个工作集可以包含若干视图,每个视图也可以属于若干工作集。同时,使用用户接口 Agent 机制将用户的共享信息和共享方法分开,使信息共享不依赖于应用程序,即使在基本应用不支持多用户信息共享的情况下也可以实现对信息的协同感知,可以较好地实现已有单用户系统的集成。

用户对共享信息的修改由接口 Agent 中专门的过程负责,当该过程接收到用户通过用户接口或从外部应用传来的修改共享信息的请求后,它首先修改共享信息的内容,并根据视图的定义通知与该信息相关的视图对象。视图对象修改视图的显示方式并通知使用该视图的相关用户,这样就保证了数据的一致性。

图 2 所示的用户接口模型允许不同地点的用户在交互时间上可以相同,也可以不同。在交互过程中增加了协同管理

功能,用户通过协作 Agent 可以很方便而灵活地加入群体之中,进行交互,同时也可以很容易地从群体中退出,中断交互操作。在交互中,通过共享窗口,采用 WYSIWIS 协同工作方式,支持用户之间的实时讨论。

在此模型中,用户结点可以实现下述 3 个层次的信息共享:

① 活动级共享,即 WYSIWIS,它要求所有用户看到的内容和形式都是相同的。当共享信息改变时,所有的显示屏幕被同时改变,因为每个视图可以属于若干工作集,用户接口代理可以方便地实现活动级共享。

② 视图级共享,所有用户看到的内容是相同的,但表现形式是不同的,即实现一种 WYGIWIG(What You Get Is What I Get)。这个功能通过复制对象和广播本地变化给所有用户来完成。例如对于相同的数据分别使用图形和表格的方法显示。通过对相同的共享信息定义不同表现形式的视图,可以实现视图级的信息共享。当共享信息被修改时,所有的定义在该信息上的视图都被更新。

③ 对象级共享,每个用户看到的内容可以是不同的,通过在不同的共享信息上定义不同的视图,可以实现对象级共享。

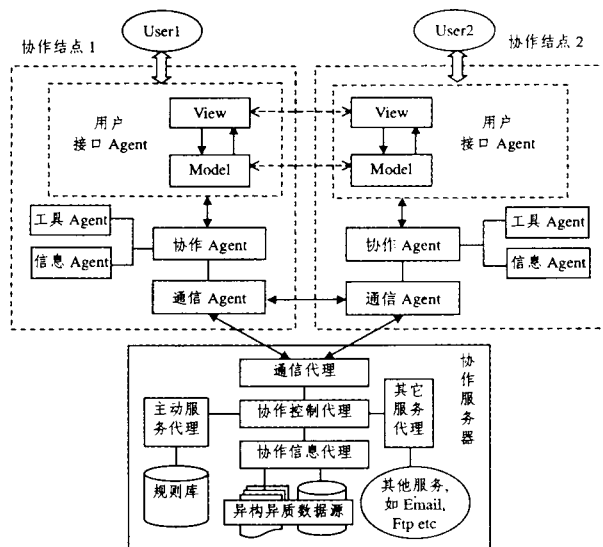


图 2 协作多用户接口模型

#### 3.3 模型特点

协作环境下的各协作成员既相互协作以共同完成同一任务,同时又相互独立地完成自己的工作。一方面,协作成员间共享信息、互相可见,并发地操作共享对象(如协同编辑中对同一文档的修改);另一方面,协作成员因阅历、层次、喜好以及习惯等的不同,将采取适合自己特点的工作方式。这就要求系统能支持成员以多种方式表达意见和显示信息。因此,协作系统中的多用户协作的输入、结果要通过协同接口来表现,被赋予了协同特征的人机接口是协作系统的重要环节,它可以充分体现协作系统中支持用户裁剪、定制和以用户为中心的设计原则。

在具体实现中,可以充分利用 XML 技术来进行用户端的定制。由于 XML 文档是平台独立的,因而对应 Model;而 XSL(eXtensible Style-sheet Language)依赖于显示的形式,因而对应 View。此结构分离了模型事件与视图事件,具有如下优点:

(1) 通过 XML/XSL 文档在文档级进行 Model-View 分离,便于进行信息的多级共享;

(2) 使用基于 ASCII XML 的标准消息格式,有助于简化协作用户间的通讯机制;

(3) 具有一个单一的数据结构(Tree)和一组相关联的操作。

### 3.4 其他相关问题

协作系统的多用户接口不仅要解决用户界面、接口组织、信息显示和操作响应等问题外。还要解决由于多用户信息共享而引起的访问冲突、一致性维护、操作共享、信息的多媒体表现与协同感知等许多新的技术问题<sup>[12]</sup>。

① 共享工作空间一致性问题 协同工作系统支持时空上分布的多个协作成员通过用户接口对共享信息或者对象进行协同操作,共享环境是通过在各个分布的协作结点上,建立一致的用户接口实现的,是实际共享环境的仿真,参与协同工作的用户可以在自己的节点上并发地操作共享信息或对象。因此,若不采用一定的并发控制机制,将很可能引发信息的不一致性。

② 群体感知问题 群体感知机制是为每个协作者建立全局的、一致的协作环境,并在此基础上提供成员与环境、成员与成员之间的信息交流途径。各用户良好的协作感知性是进行协同工作的根本前提。因为在一个协同系统中,察觉、认识和理解其它协作者的活动是人人交互、通信和协作的基本要求,用户只有在对系统全局环境、在场的所有协作者、合作任务以及共享对象有了清楚、全面的认识后,才能很好地参与多用户协作。

③ 并发控制问题 协作系统的目标是为一组用户完成一个共同的任务而提供共享环境的计算机技术支持。它允许多个用户对共同的对象(如协作文档、协同诊断系统中的医学影像等)同时进行操作。由于各用户操作的并发性和网络传输延迟的影响,如果对各用户的操作不加以控制,往往会造成各用户端上对共享对象的操作执行顺序出现不一致,从而导致各用户端的共享对象的不一致,影响了协同工作的正常进

行。

**结束语** 协作多用户接口是 CSCW 系统研究的重要内容和应用基础。本文讨论了 CSCW 中用户接口设备的多样性、用户个性化等需求,提出了一个基于 MVC 模式的协作多用户接口模型,并在我们所开发的远程协同诊断系统中得到了实现。我们将进一步深入研究基于 XML 的异构多用户端设备环境下的协作接口模型。

### 参考文献

- 1 Marsic I, Dorohonceanu B. Flexible User Interfaces For Group Collaboration[J]. International Journal of Human-Computer Interaction, 2003, 15(3): 337~360
- 2 郑庆华, 李人厚. CSCW 协作多用户接口的设计与实现[J]. 小型微型计算机系统, 1998, 19(1): 40~46
- 3 茅兵, 杜兴, 谢立, 等. 基于知识的 CSCW 人机接口模型[J]. 计算机学报, 1996, 19(3): 215~220
- 4 Dorohonceanu B, Sletterink B, Marsic I. A novel user interface for group collaboration[C/OL]. In: Proc. of the Hawai'i Intl. Conf. on System Sciences (HICSS-33), Maui, Hawaii, Jan. 2000
- 5 Smith G B. A Shared Object Layer to Support Cooperative User Interfaces[D/OL]. Lancaster University, UK, Aug. 1995
- 6 郑庆华. CSCW 理论、技术与实现[D]. [西安交通大学博士论文]. 西安, 1997
- 7 李军怀, 周明全, 耿国华. 异构环境下的 CSCW 模型研究与实现[J]. 计算机科学, 2002, 29(10): 104~106
- 8 史美林, 向勇, 杨光信. 计算机支持的协同工作理论与应用[M]. 北京: 电子工业出版社, 2000
- 9 Bentley R, Rodden T, Sawyer P, et al. Architectural Support for Cooperative Multiuser Interfaces[J]. IEEE Computer, 1994, 27(5): 37~45
- 10 茅兵, 谢立. 基于对象的协同计算模型[J]. 中国科学(E 辑), 1997, 27(6): 542~547
- 11 Bentley R, Rodden T, Sawyer P, et al. An architecture for tailoring cooperative multi-user displays[C/OL]. In: Proc. of the ACM Conf. on Computer Supported Cooperative Work (CSCW'92), Toronto, Canada, 1992. 187~194
- 12 Dourish P, Bellotti V. Awareness and Coordination in Shared Workspaces. In: Proc. of CSCW '92 [C/OL], Toronto, Canada, ACM Press, 1992. 107~114

(上接第 175 页)

(2) 波音-737, 4019, 08:03 起飞, 右转直接出航。

(3) 图-154, 4002, 长五边, 08:06 着陆。

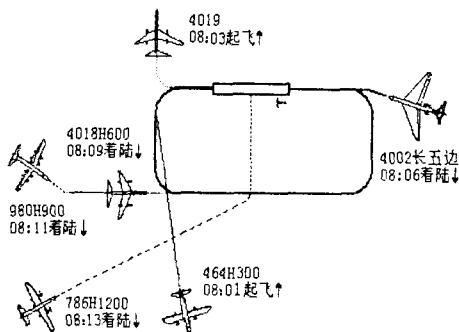


图 6 飞机指挥预案图

(4) 波音-737, 4018, 高度 600 米, 直接加入三边, 08:09 着陆。

(5) 运-8, 980, 高度 900 米, 直接加入三边, 三转弯后下降高度, 08:11 着陆。

(6) 安-26, 786, 高度 1200 米, 通场加入一边, 一边延长 30

秒, 一转弯后下降高度, 08:13 着陆。

**结束语** 应用需求推动发展, 智能决策与地理信息系统的耦合已经成为信息系统发展的一个极具影响力的趋势之一。如何将智能决策与地理信息相结合以形成有效的决策支持涉及多个学科的综合和相关领域的专业知识, 虽然人们为此进行了若干年的研究, 但仍然存在许多问题。本文以航空管制为背景, 研究了基于 GIS 的航管决策支持系统的结构、知识表示与推理问题, 实现了该系统的一个原型。原型中实现了飞行预案的自动编制和动态调整, 并利用可视化技术对飞机的起降全过程进行了动态模拟, 较好地解决了飞行指挥的主要问题。进一步完善原型系统, 使之更加符合实际需求仍需进一步的理论研究与应用实践。

### 参考文献

- 1 陈文伟. 智能决策支持系统. 北京: 清华大学出版社, 2000
- 2 朱济杰. 国际民航组织 CNS/ATM 系统的近期发展. 空中交通管理, 1997
- 3 高毅. 介绍美国交通管理系统. 空中交通管理, 1998. 3
- 4 危辉. 认知相关性与智能系统构造的系统观点. [浙江大学博士后研究工作]. 杭州, 2000