

命令通道机制

人人交互

人人协作
计算机网络

24

人人交互与协作系统中命令通道机制的实现

The Realization of Command Channel-Mechanism in a System for Human to Human Interaction and Co-operative

88-90

田友胜 侯义斌 R-J. Beun* 黄樟钦 黄进锋
(西安交通大学电脑信息研究所 西安710049)

TP393

(Eindhoven University of Technology 5600MB Eindhoven The Netherlands)*

Abstract The concept of command channel-mechanism has been introduced into the command channel-mechanism system for human to human interaction and co-operative over the Internet/Intranet. A new type of interaction and co-operative environment has been built up. This paper discusses the running mechanism and software realization of the command channel.

Keywords Interaction and co-operative, Command channel, Socket

网络的发展为计算机开辟了新的应用领域。1993年9月5日,美国宣布“国家信息基础设施(NII, National Information Infrastructure)行动计划”;1993年12月,我国国务院制定了中国国家信息化基础设施(CNII, China National Information Infrastructure)“发展纲要要点”等计划;几年前, Sun 公司就提出了“网络就是计算机”的口号;所有这些都推动了计算机网络的发展,也拓宽了计算机的应用领域。

社会化大生产需要人们有更多,更好的交流时间和交流途径。随着 Internet/Intranet 在全球范围内的普及,合作伙伴之间通过 Internet/Intranet 来进行交互与协作的需要越来越多,以 Internet/Intranet 为基础的人人交互与协作受到了人们的重视^[1,2]。

交互与协作主要分为同步和异步两种^[3]。异步方式支持非实时的交互与协作,包括电子邮件、电子公告板、新闻组、共享数据库、WWW 等。同步方式支持实时的交互与协作,可以实现 WYSIWIS (What-You-See-Is-What-I-See) 功能,包括多用户文本编辑、电子投票、网络会议、协同设计、网络讨论等。

我们研制的基于 Internet/Intranet 的命令通道机制人人交互与协作系统属于同步交互与协作系统。本文主要讨论该系统中命令通道的运行机制以及命令通道的软件实现。

1 命令通道的运行机制

基于 Internet/Intranet 的命令通道机制人人交互与协作系统采用了分布数据集中控制模型^[3]。系统把交互与协作节点分成服务节点和客户节点,在服务节点存放有协作环境数据。服务节点与各客户节点之间建立命令和数据两类通道,服务节点利用数据通道向各个客户节点发送协作环境数据。命令通道用于传送各种控制命令信息。在交互与协作组成员中,每个用户终端都保存一份协作应用程序及应用程序所用到的数据,中央服务节点负责接收和广播用户控制命令信息。当用户控制协作应用程序时,控制命令信息被发送到服务节点,服务节点把接收到的控制命令信息再广播给所有交互与协作用户。各协作用户在接收到服务节点发送过来的控制命令信息后,执行命令信息所代表的控制命令。

系统中命令通道状态变化如图1。通道建立后,命令通道处于初始状态。如果是服务节点,则命令通道由初始状态进入就绪状态。如果是非服务节点,在启动协作时要发送“下载请求”命令给服务节点,并进入下载状态。服务节点在发送完下载数据后,向请求节点发送“下载完成”命令,在收到服务节点发出的“下载完成”命令后,命令通道由下载状态进入就绪状态。如果下载不成功,则由下载状态转入初始状态。

田友胜 博士生,研究方向为广义人机交互;侯义斌 教授,博导,研究方向为广义人机交互,Internet 理论与技术;R-J. Beun 教授;黄樟钦 博士生,副教授,研究方向为多媒体技术及其在 Internet 上的应用,电子商务与安全;黄进锋 硕士生,研究方向为多媒体技术及其在 Internet 上的应用。

在命令通道就绪状态,可以利用命令通道传输控制命令信息,本地协作应用程序的控制命令信息到达后,控制命令通道将由就绪状态转变到发送状态,在发送状态,将控制命令信息按规定的格式转化,并发送给服务节点。发送完成后,命令通道再次回到就绪状态。服务节点控制命令信息到达后,控制命令通道将由就绪状态转变到接收状态,在接收状态接收控制命令信息,并将其按规定格式转化后传送给协作应用程序。接收完成后,命令通道反回就绪状态。

在退出协作时,如果该节点不是服务节点,则需发送“协作退出”命令给服务节点,并由就绪状态转变到初始状态。服务节点在接收到客户节点发送的“协作退出”命令后,将终止向该客户节点发送控制命令信息。如果是服务节点,则由就绪状态转变到退出状态。在退出状态,服务节点向各在线客户发出“退出协作”命令。各客户节点在接收到服务节点的“退出协作”命令后,做出应答,并退出协作。服务节点在收到所有客户节点的应答后,由退出状态转变到初始状态。

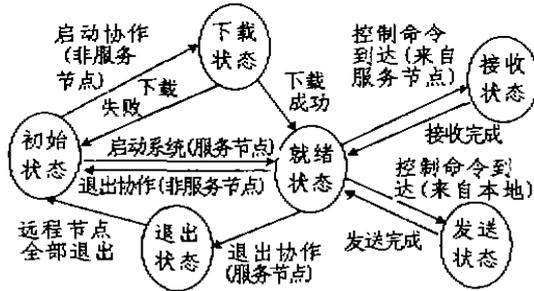


图1 命令通道状态变化

2 命令通道机制的实现

命令通道机制的功能主要通过一个通信软件实现,在通信软件中定义了控制命令信息帧的格式和命令通道机制的通信协议。

2.1 控制命令信息帧格式

由于各种控制命令信息的长度不同,划定帧边界采用了字符计数法^[4]。整个帧可分为帧头和帧体两部分,帧头由用户 ID (User ID)、控制命令类型 (Command Type)、数据长度 (Data Length) 组成;帧体由控制命令数据 (Command Data) 组成。帧结构如图2所示。

User ID	Command Type	Data Length	Command Data
用户 ID	命令类型	数据长度	控制命令数据

图2 控制命令信息帧结构图

用户 ID 字段表示所发控制命令信息用户的 ID

号,长32位。该字段表示控制命令信息的源地址,服务节点或其它客户节点可以通过查询本地的用户列表查找发出该控制命令信息的源节点地址。

命令类型 (Command Type) 字段表示该控制命令信息的命令类型,长32位。该字段表示控制命令的处理方法,每个节点都保存有一份控制命令类型表,其中包括了用户信息、下载请求、下载完成、协作退出、退出协作等控制命令,服务节点和用户节点根据该字段来决定对控制命令信息的处理。

数据长度 (Data Length) 字段表示帧中控制命令信息的长度,本字段长32位。设置该字段主要是为了方便控制命令信息数据的获取,处理程序可根据本字段的值判断帧尾的位置。

控制命令数据 (Command Data) 字段表示控制命令信息的内容,本字段是整个帧的主体部分,用于提供对用户协作应用程序的控制命令或在客户节点向服务节点连接时提供用户信息。

在软件实现中,为帧定义了一个 Comm_Frame 类,其定义如下:

```

class Comm_Frame:public CObject
{
protected
    DECLARE_DYNCREATE(CComm_Frame)
public:
    Comm_Frame();
    unsigned User_ID;
    unsigned Comm_Type;
    unsigned Data_Length;
    BYTE Comm_Data[BUFSIZE];
    virtual ~CComm_Frame();
};
    
```

2.2 软件实现

命令通道为节点之间协作应用程序控制命令的传输提供服务。命令通道机制的软件结构如图3。图中虚拟设备线程是基于 Internet/Intranet 的命令通道机制人人交互与协作系统中编写的 Windows 虚拟设备线程,用来截获本地控制命令和生成远程控制命令。

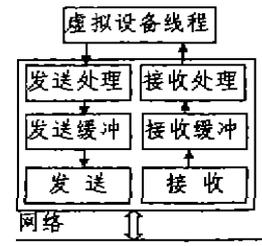


图3 命令通道机制软件结构图

命令通道主要通过接收和发送两个功能模块来实现。接收模块又分为接收、接收缓冲和接收处理三部分。接收部分负责接收服务节点发送来的控制命令信

息,并将接收到的信息放到接收缓冲区中;接收处理部分负责将接收缓冲区中的控制命令信息按帧中的控制命令类型进行处理,并将对协作应用程序的控制命令信息传送给虚拟设备线程,发送模块分为发送处理、发送缓冲和发送三部分。发送处理部分负责将虚拟设备线程截获的对协作应用程序的控制命令信息转化成规定的格式,并存放于发送缓冲区中;发送部分由发送缓冲区中取出发送数据,并发送给服务节点。

程序中通信部分用 Winsock 实现,通信采用了面向连接的 TCP Socket。为了完成发送与接收功能,我们定义了一个 Comm_Socket 类,利用它来调用发送和接收 Socket 完成控制命令信息传输的功能。Comm_Socket 类的定义如下:

```
class Comm_Socket
{
public:
    Comm_Socket(Comm_FrameDoc *pDoc);
    Comm_Socket();
    virtual ~Comm_Socket();
public:
    void Send_Comm(unsigned Comm_Type, unsigned Data_Length,
        BYTE Comm_Data[BUFSIZE]);
    void Exit();
    void Deal_Comm(Comm_Msg * comm_msg);
    void Receive_Comm();
    void Destroy();
    BOOL Init();
    CArchive * Send_Archive;
    CArchive * Receive_Archive;
    CSocketFile * Comm_File;
    Comm_Msg * Receive_Msg;
protected:
    CString Node_Addr;
    CString User_Name;
};
```

```
UINT Node_Port;
unsigned User_ID;
Comm_FrameDoc * Node_FrameDoc;
CTypedPtrList (CPtrList, Comm_Msg *) Comm_MsgList;
CTypedPtrList (CPtrList, Coll_UserInfo *) Coll_UserList;
};
```

命令通道使基于 Internet/Intranet 的命令通道机制人人交互与协作系统将组内各用户节点的协作应用程序连成一体,利用协作应用程序控制权的转移实现了 Internet/Intranet 上的交互与协作,并较成功地实现了 WYSIWIS。

结束语 基于 Internet/Intranet 的命令通道机制人人交互与协作系统引进了命令通道机制的概念,使命令与数据、音频、视频等的传输相分离,保障了协作命令的传输,降低了协作用户数对协作效果的影响,改善了协作效果,较好地实现了 WYSIWIS。

参考文献

- 1 Shah D, Candy L, Edmonds E. An investigation into supporting collaboration over the Internet. *Computer Communication*, 1998, 20: 1458~1466
- 2 Nurcan S. Analysis and design of co-operative work process: a framework. *Information and Software Technology*, 1998, 40: 143~156
- 3 Pendergast M O. A Comparative Analysis of Groupware Application Protocols. *Computer Communication Review*, 1998 (Jan.): 28~40
- 4 Tanenbaum A S. *Computer Networks*. 清华大学出版社, 1997. 2

《计算机科学》继续被编入

《中文核心期刊要目总览》2000年版

《中文核心期刊要目总览》(2000年版)编委会依据文献计量学的原理和方法,经过研究人员对相关文献的检索、计算和分析,并请学科专家鉴定,将《计算机科学》确定为自动化技术、计算机技术类的核心期刊,并编入《中文核心期刊要目总览》2000年版(即第三版)。

此次重新研制筛选中文核心期刊,被列为“国家教委人文社会科学研究‘九五’规划项目”,研制者认真总结前两次研制工作的经验,参考了大量国内外相关文献,对核心期刊的理论和方法进行了深入的研究和探讨,力求使研制方法更加科学合理,采用了中国科学院文献情报中心、中国社会科学院文献信息中心、上海图书馆、中国人民大学书报资料中心及西南信息中心等单位编制的文献数据库作为统计工具,经过严格的筛选,从我国正在出版的近万种中文期刊中确定了1568种期刊为核心期刊,《计算机科学》名列其中。

筛选和确定核心期刊的工作,是运用科学方法对各种刊物在一定时期内所刊载的论文质量和水平进行综合评价的一种科研活动,其研究工作不受外界因素影响。经过筛选所确定的各类核心期刊表是一项建立在科学研究基础上的学术成果,具有权威性,备受高等院校、期刊管理部门及广大期刊编辑部的关注和充分重视,而且有利于促进我国科技期刊的发展和公平竞争,得到业界人士好评。