

doi: 10.3969/j.issn.1006-1576.2011.11.013

一种分布式任务软件系统的远程控制方法

段慧芬, 孙丰, 王华, 刘焕敏, 顾培
(中国卫星海上测控部技术部, 江苏 江阴 214431)

摘要: 为了更好地协调航天飞行试验中的任务, 提出一种分布式任务软件系统的远程控制方法。利用软件代理技术完成不同功能的指挥显示软件模块的协调方法, 代替以往依靠岗位人员之间进行协调操作, 并在试验任务指挥监视系统中实现了代理。试验结果表明: 该方法提高了操作的可靠性和工作效率, 保证了完成任务的质量。

关键词: 远程控制; 软件代理; 指挥显示

中图分类号: TJ86 **文献标志码:** A

A Remote Control Method of Distributed Task Software System

Duan Huifen, Sun Feng, Wang Hua, Liu Huanmin, Gu Pei
(Technology Department, Satellite Maritime Tracking & Controlling Department of China, Jiangyin 214431, China)

Abstract: In order to coordinate the aerospace launching mission in a better way, a distributed software remote control method is proposed. Different display software modules of different functions can be coordinated via software agent technology, so the operators' coordinating job can be replaced, and the agent system has been implemented. The experiment result shows that this method improved the reliability and the efficiency of software operation, so the quality of the experimental task can be ensured.

Keywords: telecontrol; software agent; command and display

0 引言

航天飞行试验是一项十分庞大而又复杂的实时操作任务, 参加的设备和人员多、涉及面广、技术业务的指挥与协调复杂, 必须借助先进的试验任务指挥监视系统, 才能保证在任务实施过程中严密有序、万无一失。在任务实施中, 指挥监视系统需要协调系统中各显示工作站, 按飞行时序将不同内容、不同媒体表现形式的显示画面准确地处理、显示在不同的显示终端上, 显示系统的规模越大, 协调的难度也就越大。为此, 在试验任务指挥监视系统设计中引进软件代理技术, 通过代理解决分布在不同终端的完成不同功能的软件模块协调工作, 取消依靠岗位人员之间进行协调操作, 从而提高操作的可靠性和工作效率, 保证了完成任务的质量。

1 软件代理技术

代理一般被用来特指那些具有一定智能和自治能力, 能根据环境变化做出响应和进行相应处理的软件系统。FIPA 组织 (the foundation for intelligent physical agent) 给出代理的定义为: 代理是某一领域中基本的行为者。它具有一项或多项任务能力并对外表现为一致的和完整的执行模型而且可以与外部软件、操作人员和通信设施进行交互^[1]。

代理具有下列特点: 1) 具有自治能力, 可以自主地执行; 2) 可以通过代理通信语言, 与代理或操

作人员交流信息(可能是直接的也可能是间接的); 3) 可以感知其工作环境的变化并做出反应。

从目前来看, 代理技术的应用主要有 2 方面: 一是多个具有精神状态的实体组成联盟, 形成代理社会, 该社会的成员能够相互磋商和合作, 共同完成某项工作, 称为多代理系统; 另一方面, 代理用于信息网络系统中, 形成网络代理, 它们能在网络上进行流动, 改变网络的通讯过程, 称为流动代理。

2 代理在试验任务指挥监视系统中的应用

2.1 代理的作用

在试验任务指挥监视系统中, 笔者把指挥显示软件设计为控制体、代理和功能体 3 类对象: 控制体的实例是用户控制界面, 用于产生控制命令、收集反馈信息、监视系统运行; 代理的实例是在各功能体所在终端, 用于接收保留控制体发出的命令, 解析控制命令, 负责传达给功能体, 激活功能体相应的处理模块, 同时, 将功能体对控制的响应, 反馈给控制体; 功能体是系统中承担各具体应用功能的软件系统, 每个功能体和代理都有相同的通信接口。代理在指挥显示中承担着控制体和功能体之间命令的传递、解释、控制的功能。

2.2 代理的应用

通过对指挥显示业务的分析, 为软件代理设计

收稿日期: 2011-07-07; 修回日期: 2011-07-28

作者简介: 段慧芬(1963—), 女, 吉林人, 硕士, 高级工程师, 从事计算机应用、指挥显示技术、信息管理研究。

了 4 类命令集，分别是任务准备、监视控制、时间控制和代理通信。

任务准备包括代理环境参数的设置、任务自动流程的配置、软件接口配置和命令集维护。代理环境参数设置，用于指定代理功能体的位置和控制体的位置。任务自动流程配置，根据任务实战计划，按时序定义出控制流程，任务中不需要人工干预，由代理按流程自动控制各功能体的操作。软件接口配置，控制体在任务准备中根据数据库中对处理对象的格式描述信息，生成显示功能体的数据接口，当接口有变化时，通过代理分发给各显示功能体，保证接口变化的一致性。命令集是代理系统的规则库，当系统有新的控制功能需要扩充时，可以通过向规则库添加新的规则来实现命令集维护。

监视控制分为公共控制类、特殊控制类。公共控制类指适用于所有功能体的规则，包括系统启动、退出、系统复位、运行和状态返回等控制规则。特殊控制类指仅适用于某类功能体的控制规则，包括通信功能体的发送控制、记盘控制，数据处理类功能体的网发控制、记库控制、轨道计算控制，显示类的接口升级、画面切换等。

时间控制分为系统对时类、特征点时间应急类。系统对时类指通过对时命令将系统中各功能体的系统时间进行统一，校正各终端机器时间带来的偏差。特征点时间应急类是针对任务中特征点时间往往标志着任务阶段的开始或结束，及时、准确接收到特征点时间显得尤为重要，为此，设计了特征点时间应急机制，作为应急手段。特征点时间包括起飞时间、星(船)箭分离时间、轨返分离时间等，以确保系统能够接收到特征点时间。

代理通信包括代理与控制体的通信和代理与功能体的通信。代理与控制体的通信主要完成命令由远程向本地终端的传输，代理与功能体的通信主要完成本地终端内命令的传输。通过采用代理技术，实现了监视显示系统远程控制一体化、工作流程程序化、控制接口规范化。

3 试验任务指挥监视系统中代理的实现

3.1 代理结构

在指挥显示这个应用系统中，我们把代理设计为三层结构，系统的结构如图 1。控制体通过传输层将控制命令传送给代理，代理通过传输层接收远端的控制命令，利用规则库中的规则在解析层解析收到的控制命令，确定需要激活的功能体，通过接

口层将控制命令推送给相应的功能体。功能体将命令的执行情况，通过接口层反馈给代理，代理在解析层判断命令的反馈信息，通过传输层，发送给控制体。

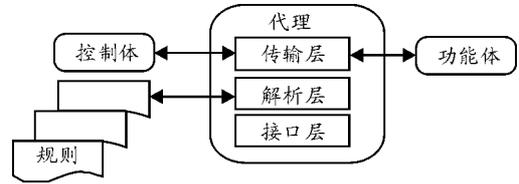


图 1 代理结构

3.2 远程控制一体化

试验任务指挥监视系统设计了一个控制体(集中控制)，4 类功能体(数据通信、数据处理、数据显示和三维仿真)。为了保证系统的可靠性，数据通信、数据处理分别设置了主、备 2 个功能体。数据显示设置了大屏幕投影显示、指挥调度显示、首长座席显示、测量船远程显示等若干数据显示功能体。系统的组成见图 2。系统中的每个功能体都通过代理接受控制中心的统一管理控制。

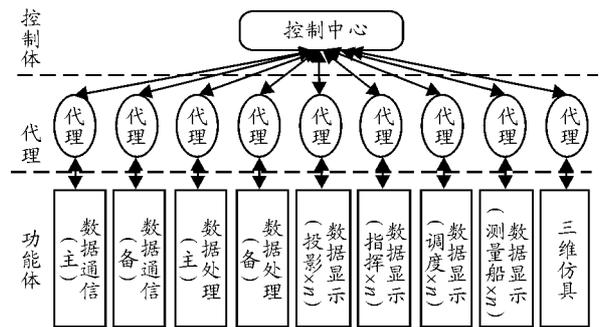


图 2 系统组成

在任务准备阶段，控制中心根据任务数据库中的配置表，生成新的接口控制文件，代理收集功能体接口文件版本，反馈给控制中心，当功能体的接口文件版本低于控制中心的接口文件版本时，控制中心将最新接口文件分发给代理，代理负责完成功能体的接口更新，保证了系统中接口的一致性。在接口有变化时，只要在控制中心将变更的接口文件版本号提高，系统中的每一个功能体在运行时都可以通过代理完成接口的更新，实现接口的统一管理，如图 3。



图 3 接口统一管理

在任务运行阶段，通过控制中心用户界面，根

据调度指挥口令向代理发送单指令, 根据代理收集到的功能体反馈信息, 再选择发送下一命令或命令重发。

控制中心与代理的通信根据命令的特点, 主要采用 2 种方法实现: TCP/IP 和 UDP 广播。TCP/IP 方式点对点通信用于对各代理软件的单独命令控制, 对某一功能体单独发送特定的命令和状态接收。UDP 方式组广播、组接收, 用于对整个控制域的通用命令广播和状态接收。代理与功能体之间采取内存映射和消息机制相结合的方法, 内存映射是将一个进程的一段虚拟地址映射为一个文件, 其它的进程可以共享该段虚拟地址, 也是一种本地进程间通信的方式, Win32 API 中共享内存(shared memory)实际就是文件映射的一种特殊情况。代理与控制中心和功能体通信的实现方法如图 4。

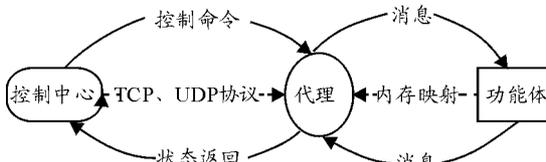


图 4 代理的通信实现

采用这种通信方式, 一方面实现代理与功能体的松耦合, 任何一方程序单独出现故障, 均不会影响到另一方程序的处理; 另一方面因为内存映射文件独立于代理和功能体而存在。所以只要一方程序未关闭, 数据便不会丢失, 这使得受控软件即使重新启动后, 仍可继续访问保存在映射文件中的数据。这种设计方法, 既实现了对功能体的控制, 又不影响功能体之间原有的逻辑关系和业务处理流程。

3.3 工作流程程序化(如图 5)

在任务运行阶段, 除了可以通过人机交互界面进行操作控制外, 还有一种自动方式。控制中心根据任务流程制定工作命令程序, 通过时间符合方式, 由程序自动将命令发送给代理, 功能体根据代理接收的命令自动响应。支持工作流程由程序控制。

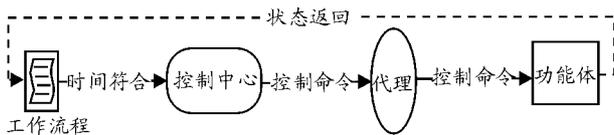


图 5 工作流程程序化

3.4 控制接口规范化

在试验任务指挥监视系统中, 与代理交互的功能体各不相同(图 2), 每个功能体的命令内容也不

尽相同, 为了便于系统的扩充和配置, 命令在规则库中的快速定位, 在设计时为代理定义了统一的接口格式、映射地址和规则库。代理在创建文件映射对象时统一使用 0xFFFFFFFF 文件句柄, 进程用 0xFFFFFFFF 来代替文件句柄(HANDLE)。

笔者为代理定义了统一接口要素, 内容包括: 校验码、命令代码、命令参数 1 及命令参数 2。校验码用于确认和识别收到的是代理命令, 避免系统误判。命令代码用于传送控制命令, 命令参数用于表示命令状态和命令执行后反馈状态。代理根据命令代码解析出命令含义, 用消息通知功能体执行相应的命令并反馈执行结果。

在软件实现中, 采用简单结构定义代理通信接口如下:

```
typedef __nogl struct SendCommand
{
    int checkout;//校验码
    int CommandCode;//命令名
    __int64 Wparam;//命令参数 1
    int Lparam;//命令参数 2
}SendCommand;
```

采用如下结构定义代理解析后的接口:

```
typedef public __value struct SqlStatus //本地用于匹配解析的结构
{
    String*CodeCn; //命令中文名
    int Wparam;// 命令名
    __int64 Lparam;//命令参数
    int Label;//标签
}SqlStatus;
```

表 1 命令集定义

命令代码	内容	LParam 低 16 位
1000	控制中心已启动就绪	
1001	启动进程	B0:1 启动 B0:0 停止
1002	程序开始	B0:1 启动 B0:0 停止
1003	系统对时	DATETIME 的 TICKS 值
1004	系统复位	
1005	要求状态返回	
1006	起飞时间 T0	DATETIME 的 TICKS 值
1007	火箭(船箭)分离点时间	DATETIME 的 TICKS 值
1008	轨返分离时间	DATETIME 的 TICKS 值
1200	网发	B0:1 启动 B0:0 停止
1201	记库	B0:1 启动 B0:0 停止
1202	发送理论弹道数据控制	B0:1 发送实际外推数据 B0:0 发送理论外推数据
1300	显示指定画面	画面编号