

支持细粒度协作的 CAD 系统在线集成研究^{*}

廖 斌^{1,2}

(湖北大学数学与计算机科学学院 武汉 430062)¹ (湖北大学应用数学湖北省重点实验室 武汉 430062)²

摘 要 研究了一种支持细粒度协作的遗产 CAD 系统的在线集成方法,提出通过中间件 agent 来封装单机 CAD 系统应用。构造了中间件 agent 的外封装器和内封装器结构,并描述其工作过程。设计了细粒度协作的通讯机制,并采用基于标准宏参命令的通讯协议来支持遗产 CAD 系统的集成。初步实现了一个集成两种遗产 CAD 应用的原型系统。

关键词 在线集成, CAD, 细粒度协作, 中间件, Agent

Research on Online Integration of CAD Systems for Fine-grained Collaboration

LIAO Bin^{1,2}

(Faculty of Mathematics and Computer Science, Hubei University, Wuhan 430062)¹

(Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan 430062)²

Abstract A method has been initially implemented to integrate legacy CAD systems for the fine-function collaboration. The method uses a middleware agent to wrap the standalone CAD systems. The structure and working processes of inside and outside encapsulations are described inside the agent. The communication mechanism of fine-function collaboration is constructed. The communication protocol based on standard macro-parametric commands is used to integrate the legacy CAD systems. A prototype system of the fine-function collaboration integrating legacy CAD applications is implemented.

Keywords Online integration, CAD, Fine-grained collaboration, Middleware, Agent

1 概述

虽然在集成 CAD/CAM 环境中可以通过交换数据文件(例如 STEP)来进行集成,并且该数据交换方法也能够用于支持协同设计,但是这种离线的集成方式的时间、空间效率比较低。其主要不足之处如下:①离线的集成方式的感知性和协作性差,当 CAD 模型被某一个用户离线修改时,其他协作用户不能了解正在进行的工作,只有在重新接收和打开文档之后才能知道;②基于数据交换的协同,其交互性和响应性差,协同粗粒度也较大,通常需要再传输整个 CAD 模型(或者相应的数据)文件。因此,人们开始对在线集成问题进行了探索,提出了若干在线集成方法。这些方法总体上可以分为两类。

第一类在线集成技术采用 CORBA、DCOM 等组件技术和中间件技术^[1]。例如,CFACA 作为一个组件框架用于基于特征的设计和过程规划,实现 CAD/CAM 集成^[2]。CFACA 的关键有两点:一是封装 ACIS 几何造型库;二是开发基于 ACIS 的新应用组件,这些组件具有相同核心接口。但是,CFACA 没有讨论如何集成现有 CAD 应用的问题。和 CFACA 相似,PRE-RMI 框架也没有考虑对现有 CAD 应用进行集成的问题^[3]。在 MIT 的 CAD 实验室提出的 DOME 框架中,提到了利用 CORBA 作为一个基础设施来集成 DOME 模块和现有应用^[4]。DOME 文献表明 DOME 模块是基于以前开发的 OME 库,但是该文献却没有说明如何集成现有应用,也没有给出一个集成现有应用的例子,以至于人们无法进行

进一步的研判。在一个基于 agent 和 CORBA 的应用集成平台中,Chan 明确地指出需要在该平台中集成现有应用^[5]。为此,Chan 提出了一个通用结构用于封装遗产系统。该结构中有两类封装器:外封装器(Outside Encapsulation)和内封装器(Inside Encapsulation)。Chan 对外封装器进行了清晰的论述,即外封装器靠近 ORB(对象请求代理),用于在 ORB 和内封装器之间传递数据和请求调用,负责把客户端的操作请求翻译成为方法来调用。在对外封装器做了很好的介绍以后,后文却没有对内封装器进行任何解释。外封装器可以按照底层互操作协议(例如 CORBA 等协议)来构造与 ORB 的接口。这样,构造外封装器的时候,通常有一个相对一般化的参考标准。相比之下,内封装器却是千差万别的,需要对各种 CAD 应用的专有特性进行剖析。这就是 Chan 没有对内封装器进行具体解释和论述的原因。

第二类在线集成方法没有采用组件技术,而是按照 Chan 对封装器所进行的分类方法,因此可以在很大程度上可以归为内封装方法。PACT(Palo Alto Collaborative Testbed)项目的相关工作就是这方面的经典案例^[6]。PACT 采用交互 agent 来封装和集成四个子系,即 NVisage, DMD, Next-Cut, Designworld。按照 PACT 文献,一个 agent 包括一个 CAD 系统和一个封装器。但是,该 agent 的具体实现细节没有在文献中给出。其中原因在于:①PACT 的主要目的在于知识共享和数据共享,agent 不是研究重点;②PACT 项目可以在一定程度上得到这些子系统的源代码,因为这些子系统不是商品化专有系统。许多研究借鉴了 PACT 中 agent 概念,但是

^{*}国家自然科学基金(60510627)资助项目。廖 斌 博士,讲师,主要研究方向为 CSCW 和协同 CAD。

却没有仔细考虑在没有源代码的情况下(例如专有 CAD 系统),如何构造 agent 来集成遗产 CAD 系统。

本文研究的目的是不是简单地开发一个分布 CAD 或者协同 CAD 应用系统(尽管需要一个原型系统来验证相关思路),而是探索一种遗产 CAD 系统的在线集成方法,从而支持细粒度协作的 CAD 应用。

2 在线集成器

本文研究了一种适合单机 CAD 系统应用的灵活、开放的在线集成方法,可以形象地称之为集成器,如图 1 所示。

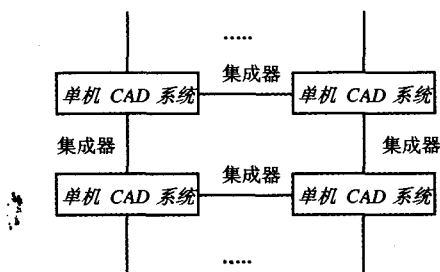


图 1 单机 CAD 系统的在线集成

如图 2 所示,集成器可以细化为三层体系结构,即架构层、中间件层、应用层。其中,架构层包括网络环境、计算机硬件和操作系统。中间件层包括基于服务和设备的远程过程调用,作为客户端应用与服务器端应用之间的互连接口。在应用层中,通过中间件 Agent 将单机 CAD 系统应用封装到客户端,而客户端的感知服务(包括群感知、任务管理、协调、组广播)则由服务器端提供。

与传统的两层体系结构(应用层直接与架构层相连接,服务器端与客户端都需要处理底层通讯机制)不同,在集成器的三层体系结构中,底层的通讯机制是由中间件层来处理。

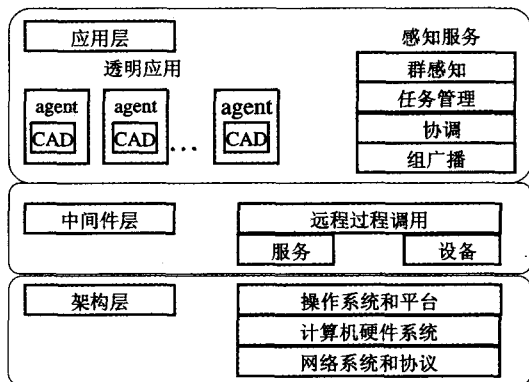


图 2 集成器的三层结构

限于篇幅,以下将主要讨论集成器的核心部件的实现,即中间件 agent 结构及其工作过程,以及通讯机制和通讯协议。

3 中间件 Agent

如图 3 所示,中间件 agent 通过外封装器和内封装器来封装单机 CAD 系统的应用。此种 agent 构成方式可以根据不同的计算机软件、硬件环境来集成遗产 CAD 应用。下面将分别讨论外封装器和内封装器的结构和功能。

3.1 外封装器

外封装器是内封装器和中间件之间联系的纽带,主要为

内封装器意图实现的协同功能和中间件的通讯功能提供接口。如图 4 所示,外封装器包含了来自于内封装器的操作请求,并将其发送到代理对象;同时,也通过代理对象接收界面感知,并将其发送到内封装器。

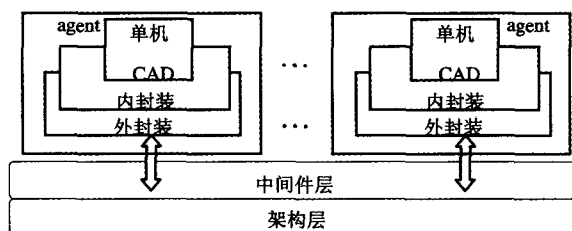


图 3 中间件 agent 结构

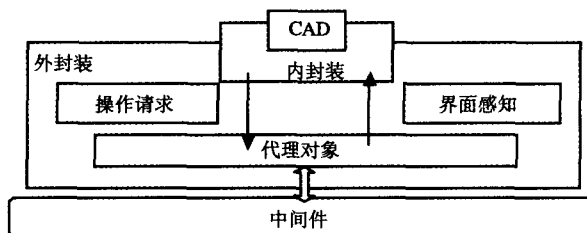


图 4 外封装器结构

一个遗产 CAD 系统即使没有中间件接口,仍然能够通过外封装器与中间件通信。这样就使得各类遗产 CAD 系统都能集成到与任何基于中间件的分布式应用中。

3.2 内封装器

内封装器直接与遗产 CAD 系统交互,是实现细粒度协作的 CAD 协同设计的最为重要的部分。如图 5 所示,内封装器采用的是一种半自动软件 Agent 结构,来封装遗产 CAD 系统。主要部件包括交互监视器、命令监视器、协作控制器、模型监视器。

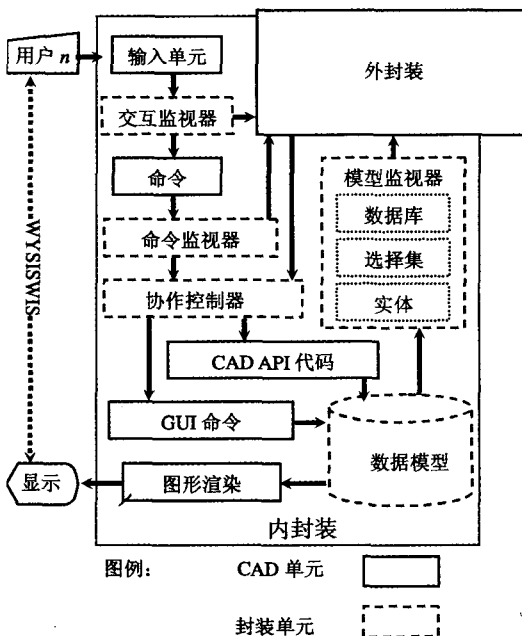


图 5 内封装器结构

• 交互监视器用于监测遗产 CAD 系统的交互事件,截取底层的用户交互信息,如键盘和鼠标事件。与现有典型协作

工具不同,agent 并不完全依赖这些底层信息。这些底层信息继续被递交给遗产 CAD 系统的命令解释器,形成高层语义命令。

- 命令监视器用于监测遗产 CAD 系统的命令解释器,获取直接与语义命令相关的高层信息,并递交给协作控制器。典型信息有以下几类:①命令名;②命令状态,包括准备开始执行、执行结束、被取消、不能执行;③文件 I/O 信息;④Undo/Redo 信息。

- 协作控制器用于协调来自命令监视器的本地信息和来自网络的远程信息,决定如何调用内建可执行命令、或者用户化 API、或者两者的组合。一般而言,遗产 CAD 系统已经内建一套常用的可执行命令,同时又提供了不同能力的 API 用于满足用户化要求。当标准宏参命令不能直接转本机命令时,这种用户化 API 功能为实现标准宏参命令的语义功能提供了支持。

- 模型监视器用于监测遗产 CAD 系统中 CAD 模型的状态。所监测的数据粒度可以是整个数据库,或者选择集,或者单个实体。模型监视器可以通过上述数据状态的典型变化来推算并规整为最基本的语义命令及其组合形式,可以有效减少与标准宏参命令所进行的一对一转换的工作量,这在某些协作工具研究和开发中是非常有利的。

4 通讯机制

4.1 分层通讯模型

进程间通讯是执行远程 CAD 命令和 API 调用从而实现细粒度协同工作的基础。为了能够支持遗产 CAD 系统应用进程之间直接和明确地通讯连接,在线集成器采用了一种分层通讯模型。

国际标准化组织制定的开放系统互连参考模型为七层,实际得到广泛应用的互联网协议族模型为四层,在线集成器的通讯模型则以互联网协议族模型为基础,大体上分为三层,即细粒度功能协作层、协作支持层和网络通讯层,如图 6 所示。

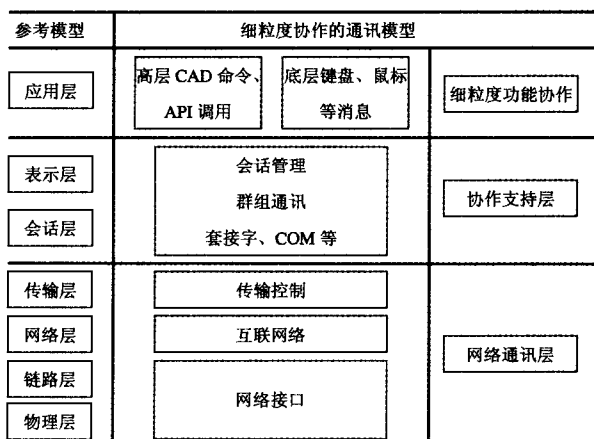


图 6 分层通讯模型

- 在细粒度功能协作层(相应于参考模型的应用层),遗产 CAD 系统进程通过交换高层语义(如 CAD 命令、API 调用)和底层消息(如鼠标、键盘事件消息)进行细粒度的功能协作。

- 协作支持层(相应于参考模型的表示层和会话层)则负责协同设计的群组通讯和会话管理等支持功能,并通过套接

字接口、COM 接口等与网络通讯层建立联系。

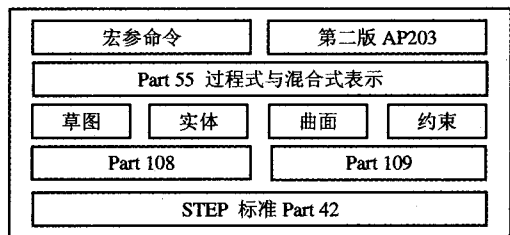
- 在网络通讯层(相应于参考模型的物理层、链路层、网络层和传输层),采用应用了互联网模型的网络接口层、互联网网络层(即 IP 层)和传输控制层(即 TCP 层)的功能。

4.2 基于标准中性宏参命令的通讯协议

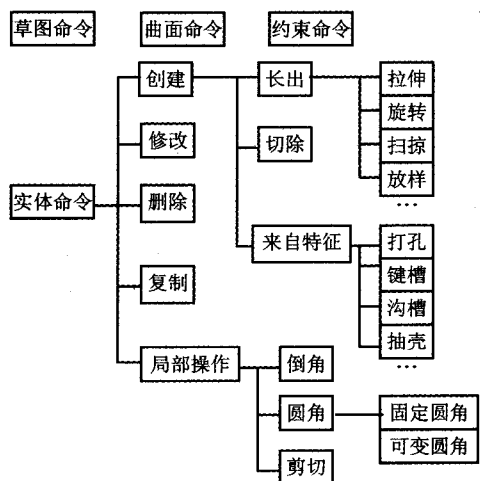
在分布式环境中,通过交换信息来实现协同工作,各个站点必需遵守相应的规则,即面向协同设计的通讯协议。除了用于协作任务、会话管理、协同感知、一致性维护和并发控制等之外,该协议的一个很重要的组成方面是如何在各个站点之间进行细粒度的设计意图交互。这就涉及到异构问题及其对应的互操作解决方案。

CORBA 等协议的主要贡献是在底层完成了方法调用和方法实现之间的映射(当然 CORBA 等协议则又要基于更底层 TCP 协议来解决计算机网络等平台之间的互操作)。为了进一步实现高层(应用语义层)的互操作,就需要对 CAD 核心 API 进行标准化,即如何调用和如何实现方法。这方面的主要工作有 AIS、SDAI、Djinn API^[7]和 OMG 的 CAD Services。但是到目前为止,还没有一种 API 标准能够把主流 CAD 系统统一起来。

尽管本文重点内容不是针对互操作问题,但是考虑到对异构 CAD 系统的集成及其互操作问题,在通讯协议中采用了 KASIT 向 ISO 建议起草的标准中性宏参 CAD 命令。标准宏参命令在 STEP 标准体系中处在 STEP Part 55、Part 108、Part 109 层面之上,大致与第二版 AP203 平行,如图 7(a)所示。通过对现有主流 CAD 系统命令的分析,中性宏参命令被分为草图命令、实体命令、曲面命令和约束命令,如图 7(b)所示。



(a) 相关标准体系



(b) 命令示例

图 7 标准中性宏参命令

通讯协议中的宏参命令与标准中性宏参 CAD 命令有几

个不同之处:

- 前者侧重于实时在线的命令交换,后者侧重于整个模型的交换。

- 协议中的宏参命令可以将一个小的命令集从一个站点传到其他的协同站点,从而可以达到很小的协同粒度,而标准宏参命令则只能将整个文件进行很粗粒度的信息共享和模型交换。

- 协议中的宏参命令以一种动态交互的、不可预知的方式交换,而标准宏参命令是以一个模型完整的造型步骤序列来交换。因此在协议中使用的宏参命令进行协同通讯的时候必须与并发控制协议配合使用。

5 原型系统

不同媒体对数据传送的可靠性有不同的要求。在多媒体视频、音频传输过程中,媒体数据的相关性小,数据丢失或者失序不会对协同工作产生严重或者长期影响。但是在本文中,依靠通讯协议中标准宏参命令及其次序来保证的各个站点上三维设计意图的一致性,必须建立可靠性较强的、全局有序的组通讯机制。目前版本的原型系统采用经典的、基于客户/服务器的、多个点到点的通讯转发模式。其优点是通讯转发服务器端程序算法实现简单,可靠性较强;缺点是需要为每一个客户端重复发送数据,适合数目不太多的协同设计小组。在协同控制方面,目前版本的原型系统采用了与上述组通讯模式相适应的发言权控制方法。

通过分别对三维 SolidWorks 和三维 MDT 进行基于中间件 agent 的封装,本文初步实现了一个支持细粒度协作的原型系统。系统运行的流程如下:

- ①启动通讯转发服务器进程,监听客户端请求。当有客户端向服务器端发出加入本次协同时,服务器会在自己的列表中显示出该客户端加入。

- ②客户端在本地机上生成请求建立零件的字符串命令。

- ③客户端将字符串命令发送到通讯转发服务器。

- ④当多个客户端同时发出命令时,将其存放于消息队列中,并不立即作用到数据库上,而是由通讯转发服务器做出仲裁,决定是否转发。

- ⑤通讯转发服务器对各操作消息进行排队,形成一致的操作序列。

- ⑥通讯转发服务器将操作序列返回到各个客户端,并对复制式的数据库拷贝进行相同的操作。

- ⑦当客户端请求加入到发言讨论组时,将由服务器进行仲裁。如果服务器批准了本次请求,则会产生一个用于与此客户端进行数据传输的线程。

如图 8 所示,在上述系统运行的流程下,两个 agent for SolidWorks 和三个 agent for MDT 在线集成协作情形。其中,采用标准宏参命令作为通讯协议,初步实现 SolidWorks 和 MDT 在协同工作时的互操作。

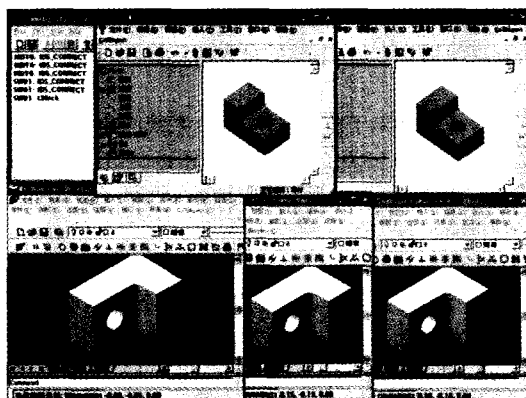


图 8 原形系统实例

结束语 本文在分析了现有集成技术的基础上,对基于中间件 agent 的在线集成问题进行了探索,研究了 CAD 系统在线集成的方法、中间件 agent 的基本结构以及相应的通讯机制和通讯协议,分别以三维 SolidWorks 和三维 MDT 零件为实验对象,进行了细粒度的协同设计。以下两个方面值得进一步探索:①如何在协同工作环境中集成更多的遗产 CAD 应用并且保证互操作;②组通讯模式对协同设计环境有直接影响,如何构造其他类型组通讯模式下的细粒度协作。

参考文献

- 1 Urban S, Dietrich S, Saxena A, et al. Interconnection of distributed components: an overview of current middleware solutions. ASME Transactions Journal of Computing & Information Science in Engineering, 2001, 1(1): 23~31
- 2 Liu X. CFACA: component framework for feature based design and process planning. Computer-aided Design, 2000, 32(7): 397~408
- 3 Gerhard J, Rosen D, Allen J, et al. A distributed product realization environment for design and manufacturing. ASME Journal of Computing and Information Science in Engineering, 2001, 1(3): 235~244
- 4 Pahng F, Senin N, Wallace D. Distributed object-based modeling and evaluation of design problems. Computer-aided Design, 1998, 30(6): 411~423
- 5 Chan F, Zhang J, Li P. Agent- and CORBA-based application integration platform for an agile manufacturing environment. International Journal of Advanced Manufacturing Technology, 2003, 21(6): 460~468
- 6 Cutkosky M, Engelmores R, Fikes R, et al. PACT: an experiment in integrating concurrent engineering systems. IEEE-Computer, 1993, 26(1): 28~37
- 7 Armstrong C, Bowyer A, Cameron S, et al. Djinn-specification & report. Information Geometers Ltd, 2000. 410