

## 一种基于 MVC 模式和插件的触觉交互软件体系结构<sup>①</sup>

刘 佳<sup>②</sup> \* \*\* 宋爱国<sup>\*\*③</sup>

(\* 南京信息工程大学信息与控制学院 南京 210044)

(\*\* 东南大学仪器科学与工程学院远程测控技术江苏省重点实验室 南京 210096)

**摘 要** 基于对触觉交互软件特点的分析,针对目前触觉交互软件应用中的突出问题给出了一种新的可扩展的触觉交互软件体系结构。该体系结构基于模型-视图-控制器(MVC)模式和插件,能够使系统进行动态扩展。在此基础上结合实例给出了三种扩展机制,分别是基于类型封装的模型扩展,基于状态转换的控制器扩展和基于发布-订阅的视图扩展。通过这些机制,开发者可以快速方便地扩展软件功能,提高开发效率。该体系对于实现触觉交互软件的标准化具有重要的价值。

**关键词** 触觉交互,可扩展性,软件体系,模型-视图-控制器(MVC),插件

## 0 引 言

触觉交互(haptic interaction)是人机交互领域的最新技术。交互式远程操作机器人技术和虚拟现实技术的广泛应用,将需要大量的触觉交互系统。触觉交互系统可以将虚拟物体的触觉信息(如柔性、表面纹理、温度等)反馈给操作人员,让操作人员操纵触摸虚拟物体,从而使他们可以准确地判断和辨别物体的物理特性和分类<sup>[1,2]</sup>。

随着触觉交互应用领域的不断扩大,对其软件体系结构设计的要求也越来越高。触觉交互软件必须具有确切的层次定位和较强的可扩展性,并能提供高效率的处理机制。目前国内外可用于触觉交互的软件体系结构主要有以下几种。一是基于面向构件的软件体系结构,这种体系将虚拟仪器划分为功能相对独立的几个构件,利用构件组装技术完成软件组装,初步实现了设计复用和过程复用,但其缺点是构件的粒度较粗,没有考虑对交互监控的支持,并且依赖于组件技术进行扩展<sup>[3]</sup>。二是用户界面-模型体系结构,这种体系对模型-视图-控制器(model-view-controller, MVC)模式进行简化,将仪器的操作与算法逻辑分离,为软件标准化提供基础,但这种层次化体系结构重点在于用户界面的标准化,对软硬件资源的标准化考虑不足<sup>[4]</sup>。三是采用了开放的体系

结构,这种结构提供了设备控制台和软件控件两类扩展点,能够实现对任意硬件设备的支持和任意的软件功能,但其缺点是依赖于 eclipse 框架,不能基于触觉交互的特点设计对应的扩展机制<sup>[5]</sup>。chai3D 是目前应用较为广泛的力触觉软件开源框架,支持手控器等多种商业化力触觉装置,并可以扩展支持自主开发的力触觉装置,其缺点是基于应用程序编程接口(API)的开发方式难度较大,并且重点在于对触觉装置等硬件资源的扩展,对算法逻辑的扩展支持较少。由以上分析可见,目前触觉交互软件体系设计的主要问题是:标准化研究较少,缺乏完善的软件体系结构;现有体系结构依赖于特定的组件或工具,这在一定程度上限制了软件的通用化;开发者面临标准化与特殊化的两难选择。软件的体系设计若提供最全面的功能而不考虑特定应用领域需求,不仅增加触觉交互系统的成本,而且不易扩展特定功能;若为每个应用领域使用一种定制的基础设施,则会失去单一解决方案的标准性和集成性,这些非标准化的实现阻碍了触觉交互技术的应用推广。基于这种认识,本文研究了触觉交互软件的特点和应用需求,以本实验室研制的一种触觉交互软件为例,提出了一种可扩展的体系结构。该结构设计基于 MVC 模式,将数据结构、数据的交互处理、前台界面相分离,通过插件管理允许在运行时增减和替换插件,从而可提高系统的灵活性。该体系对实现触觉

① 863 计划(2006AA01Z329, 2006AA04Z246)和国家自然科学基金(60775057, 60675047)资助项目。

② 女, 1981 年生, 博士, 讲师; 研究方向: 虚拟现实人机交互技术。

③ 通讯作者, E-mail: a.g.song@seu.edu.cn

(收稿日期: 2008-10-30)

交互软件的标准化有重要价值。

## 1 触觉交互软件体系结构设计

### 1.1 触觉交互软件标准化需求分析

如图 1 所示,触觉交互系统包括操作者、触觉交互装置、仿真计算机等部分。当操作者与虚拟环境交互时,一方面通过眼睛观察到虚拟场景的视觉信息,另一方面通过触觉交互装置与该环境交互。操作者的手通常被映射为一个虚拟手,操作者通过触觉再现装置控制虚拟环境中的虚拟手与虚拟物体相互作用,虚拟环境中物体的触觉信息又通过触觉交互装置反馈给操作者,以达到视觉和触觉的协调一致。

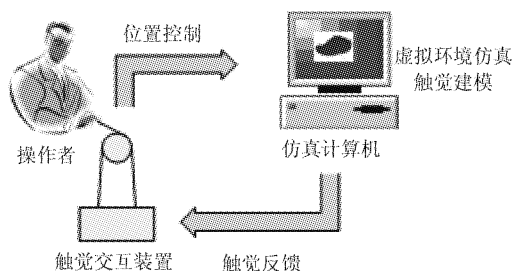


图 1 触觉交互系统

系统软件部分的功能包括建立各种虚拟物体和虚拟手的模型,描述它们的几何、物理属性。在系统交互过程中计算操作产生的变形,通过图形显示系统实时显示三维图形场景,并与触觉显示同步,从

而给操作者提供逼真的操作感受<sup>[6]</sup>。装置控制、交互信息采集、虚拟环境建模、物理仿真、虚拟触觉计算、虚拟场景显示软件的功能,可以抽象为三类最具代表性的需求,即资源管理、触觉感知、交互监控。

资源管理用于实现和软件基础数据相关的需求,包括对触觉交互装置的控制、实时数据采集虚拟物体模型的加载和规格化、实时数据持久化等。由于目前有多种软硬件技术和工具用于触觉交互系统,这类需求的特征是提高系统的扩展性,以满足不同的硬件和软件接口。

触觉感知需求包括虚拟触觉计算、命令序列检测和交互响应、虚拟物体的渲染、虚拟场景的刷新等。这类需求的特征是响应系统内部的命令并完成系统的输出,为最终用户提供友好的处理界面和流畅的感知效果。

交互监控是最复杂的需求,用于管理系统的输入和内部交互,包括交互事件时效检查、命令序列发送和持久化机制、虚拟物体的碰撞检测和变形处理、处理频率监控和调整等功能。这类需求的特征是对处理逻辑算法等进行抽象,并能对数据流进行控制和优化,使整个系统高效稳定运行。

为了达到可扩展、可配置的软件设计目标,体系结构不宜设计成涵盖各种需求的大系统,而应该是具有良好接口的介于应用层和触觉再现硬件装置之间的软件平台,并允许应用开发设计人员方便地针对以上三类需求扩展系统功能。表 1 给出了在系统设计中要解决的主要需求,以及它们各自的扩展机制,也举例了其扩展的特征。

表 1 系统需求

| 需求   | 扩展机制                       | 扩展特征                                                     |
|------|----------------------------|----------------------------------------------------------|
| 资源管理 | 标准化的数据结构<br>基于泛型的序列化       | 可支持 3DMAX 等多种工具建立的模型<br>数据可保存到多种文件或数据库                   |
| 触觉感知 | 标准化的事件定义<br>事件订阅/通知定义部分可扩展 | 实现交互操作的记录和重播功能<br>可以为一种事件实现多种显示界面                        |
| 交互监控 | 配置文件<br>交互/事件转换和时效检测       | 软件的功能模块可配置<br>动态改变物理仿真算法<br>可调整数据采集、界面刷新等频率,以及<br>频率的自适应 |

### 1.2 体系结构设计

设计一个可扩展、可配置、标准化的触觉交互软件体系结构的关键是刻画对交互的处理机制,提供允许动态增加、替换交互处理功能(功能性需求)的基础设施,并引入非功能方面的处理,如时效检测和

频率调整等。针对以上需求,本文提出了一种将 MVC 模式与插件相结合的体系结构。MVC 模式基本思想是数据、显示和处理相分离,开发具有伸缩性、便于扩展、便于维护的应用软件<sup>[7]</sup>。插件是指一种遵循一定规范的应用接口编写出来的程序,从基

本的接口派生出实现各种具体功能的插件,通过插件管理器动态的加载和释放,在插件中调用具体的资源<sup>[8]</sup>。

如图 2 所示,该体系结构可划分为基础结构和应用结构两部分。其中基础结构包括插件框架、配

置文件、异常处理、数据访问等支持系统运行的基础功能。系统的可扩展性基于插件管理器实现,在初始化或应用请求时,插件管理器根据配置和请求信息动态地增加、去除、替换具体的插件。

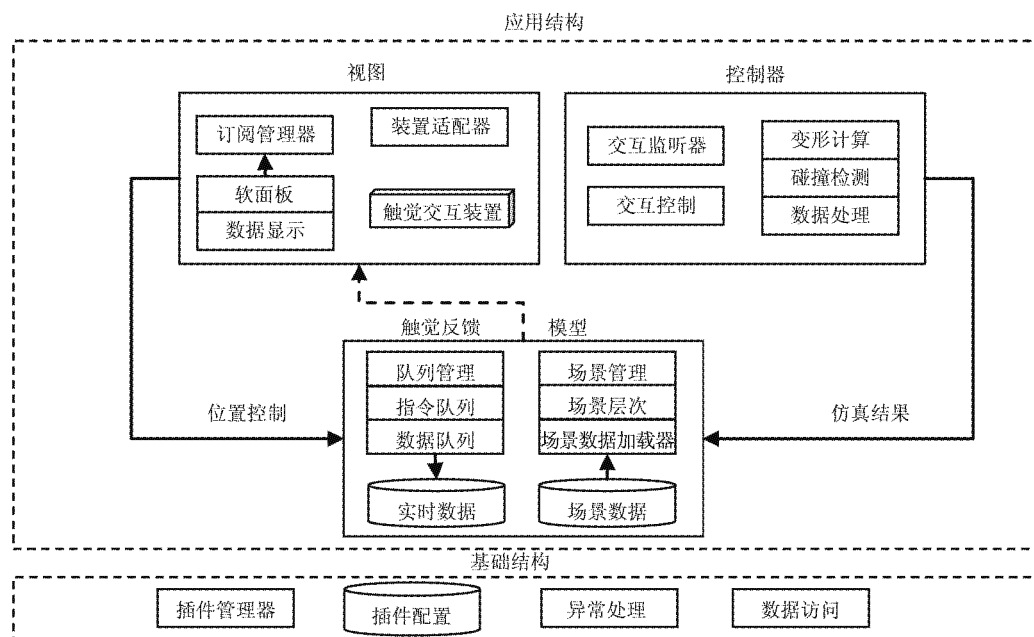


图 2 软件体系结构

应用结构采用 MVC 模式设计。模型对应于资源管理需求,用于管理和维护数据,并提供数据查询和操作的接口,当数据改变时通知相应的视图进行刷新。触觉交互系统中的数据可划分为实时数据和虚拟场景数据两部分。实时数据指系统运行交互产生的数据,按数据类型用多个循环队列进行管理。虚拟场景数据指构建虚拟环境所需要的模型数据,可划分为物体-面-顶点等多个层次,按照配置进行装载和管理。视图对应于触觉感知需求,用于展现应用数据,以图形、文件、输出设备等方式将模型的状态显示给用户。在视图中用订阅机制将操作和用户面板实现隔离,使界面组件容易修改和复用;并用适配器实现对力触觉硬件装置的扩展,将当前位置等信息提交给模型。控制器对应于交互监控需求,用于管理视图和模型的交互。基于有限状态机跟踪和处理用户操作事件,调用模型的相应接口使用仿真结果更新当前场景,自动实现视图的实时更新和触觉反馈。资源、场景、界面组件、事件、消息、策略描述等对象都在插件配置文件中定义,将对象间固定的引用方式变成灵活的连接方式,降低了对

象之间的依赖性,从而提高了应用结构的可扩展性和可复用性。

## 2 可扩展的触觉交互软件体系的实现

本实验室自行研制了一套小型触觉交互装置,并在此基础上构建了触觉交互系统,可给操作者同时提供视觉、触觉的实时感知交互信息,如图 3 所示。触觉交互装置负责采集操作者的运动,操作者可以通过该装置用手指直接触摸虚拟环境中物体并感知其表面的刚度。仿真计算机配置为 Pentium 4 2.80GHz,1GB 内存,仿真软件运行平台为 Windows XP,编程语言为 VC++,图形模块基于 OpenGL 和 3DS MAX 开发。

软件体系结构中的基础结构用于实现相对稳定的非功能需求,一般不需要扩展。对于和应用场景有关、易变的功能性需求,主要是在应用结构中进行扩展实现。应用结构由模型、控制器、视图三部分组成,所以通过对这三部分的扩展即可实现软件功能的全方位扩展。

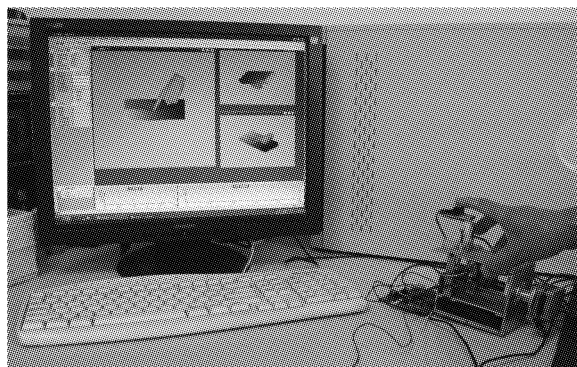


图 3 基于小型触觉交互装置的触觉交互系统

## 2.1 基于类型封装的模型扩展机制

模型的功能是封装数据和对数据的操作。本触觉交互系统中的数据可以分为两类:虚拟场景和实时数据。虚拟场景数据量相对固定,在系统初始化时加载;实时数据包括采样数据和设备控制指令,数据量随系统运行而增长。因此分别采用层次结构和循环队列结构保存虚拟场景和实时数据。对模型的扩展包括数据类型的扩展和数据操作的扩展。

对数据类型的扩展主要基于虚拟场景实现,包括场景、模型、物体与面片等多个层次。每个层次都定义了一批数据类型用于存储属性信息,包括场景的组织结构信息(父节点或子节点的句柄)、模型描述信息(包围盒和坐标位置)、特征属性信息(变换矩阵、材质等)。数据类型和数据来源都通过插件配置文件进行描述,因此这种层次结构模型可以基于统一的管理维护方式将各种模型元素有机地组织起来。如需增加新的虚拟物体,只需在配置文件中增加文件路径、物体类别、对应菜单项等描述信息,即可实现灵活修改虚拟场景,并自动更改用户界面的目标。

对数据的操作可以抽象为数据类型、访问权限和处理逻辑。通过对操作的扩展,不仅能实现数据的动态加载,而且可以改变对数据的处理过程,如更改包围盒形状、调整算法精度、抽取采样数据等。由于定义了统一的操作接口,对操作的扩展只需要重载对数据的遍历方法,并在处理循环中给出自己的状态,从而使虚拟场景中的交互行为得到扩展。

## 2.2 基于状态转换的控制器扩展机制

控制器用于管理视图和模型的交互。扩展方法是抽象出表示一类模型元素处理规范的控制器接口,然后通过派生出不同的控制器,即可在模型定义不变的前提下扩展对模型元素的处理方式。

我们把软件的运行过程划分为不同的状态,在

状态变换时加载不同的控制器,以增强软件的扩展性。本文描述的触觉交互软件可以划分为装置调整、实验数据采集、虚拟场景调整、虚拟手位置调整四种状态。在这些状态下的不同输入,如触觉交互装置的数据采集、用户在界面上的鼠标操作等,都以队列方式缓存,并根据当前状态和配置加载对应的控制器。在增强扩展性的同时,也提高了处理效率。

采用状态转换还可以协调工作频率。由于在虚拟物体的变形过程中,只对受到影响的点进行计算,计算开销和变形量之间存在平方关系。随着变形量的增加,较精细的虚拟物体会产生计算滞后问题,所以需要按照繁忙程度对工作频率进行自适应调整。我们把虚拟场景调整状态中的物体变形处理划分为四个状态:分离、碰撞检测、变形、分离检测。在虚拟手接近虚拟物体时,从分离状态进入碰撞检测状态,为避免在快速移动时检测不到碰撞,以较高的频率进行碰撞检测。发生碰撞后进入变形状态,由于变形是一个累积的过程,可适当降低处理频率,同时可以加载并改变当前的变形算法。在变形减少,接近恢复到虚拟物体原始形状时进入分离检测状态,也需以较高的频率进行处理。

## 2.3 基于发布-订阅的视图扩展机制

视图用于封装模型的展现方式。本文描述的触觉交互软件结构以模型为核心,利用插件完成视图的动态加载。由于一种数据可能同时具有多种展现方式,使用发布-订阅模式定义模型和视图之间一对多引用关系。每个视图被加载后,都按照配置在模型中注册对应的订阅接口,当模型中的一类数据对象发生变更时,所有的订阅视图都得到通知并自动刷新。

这种扩展机制主要解决了三类问题:(1)数据发布的开放性。在操作者移动装置、进行键盘/鼠标操作时,都对应于虚拟手的位置的调整,即为相同的数据提供多种数据发布方式。(2)后台数据和前台展现的解耦合。虚拟手与虚拟物体相接触时,根据虚拟物体表面接触点的刚度计算出触觉再现装置中弹性梁的有效长度,并通过位移传感器获得的操作者手指按压的位移值来计算虚拟物体表面的变形。触觉装置视图和虚拟场景视图都订阅了这些数据,在数据被更新时得到消息通知,控制电机转动调整装置,并刷新图形界面,以达到触觉视觉协调一致。(3)有效管理视图扩展。由于发布-订阅是基于配置方式,新增的虚拟物体柔软度设置、虚拟环境视角调整等操作都只用更改配置,而不用更改已有的实现。

## 2.4 应用及效果

在没有实现扩展插件的情况下,系统软件只具有基本的参数设置功能,在场景中没有虚拟物体,界面如图 4 所示。

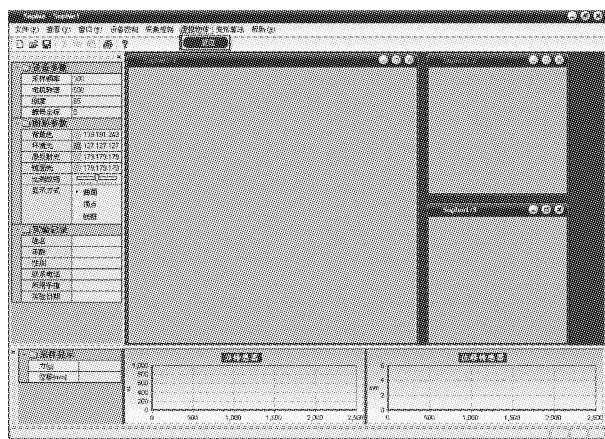


图 4 初始软件界面

在实现手、肝脏等虚拟物体、变形算法等扩展插件,并按照目录规范完成部署后,系统自动加载插件并组装界面,如图 5 所示。扩展前后软件运行稳定。

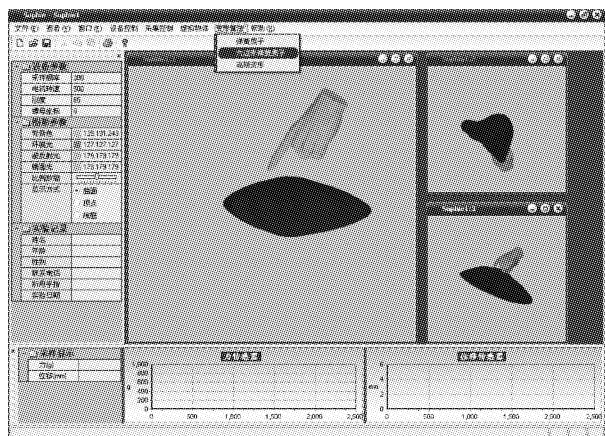


图 5 实现扩展后的软件界面

采用该体系结构的主要优点在于:(1)加强了软件的复用性,只用修改配置文件就能构造出不同功能的系统。(2)有利于模块化的开发方式,插件可以独立编译,如果需要进行功能扩展,只需要增加或修改插件,而不用重新编译整套软件。(3)各个插件之间隐藏了内部细节,仅通过接口进行交互,并允许在运行时动态增减和替换插件,容易进行部署和升级,提高了软件的灵活性。(4)能在基础结构中以统一的方式进行异常处理,增强了软件的可靠性。

## 3 结 论

本文结合实例提出了一种将 MVC 模式与插件相结合的触觉交互软件体系结构,并在此基础上讨论了三种扩展机制。扩展机制利用统一的方式加载和释放插件,并在派生插件中实现各种具体功能,调用具体的资源。针对目前绘制对象描述能力不足的问题,允许应用开发者扩展层次化模型元素。基于状态转换的扩展机制不仅允许改变不同状态下系统的行为,适合于实现可扩展的处理算法,而且可以调整工作频率,优化对系统资源的使用。基于订阅-发布实现视图扩展的机制可以使后台数据和前台界面解耦合,便于增加数据展现方式,以达到触觉视觉的协调。

## 参考文献

- [1] Taylor R H, Stoianovici D. Medical robot in computer-integrated surgery. *IEEE Trans on Robotics and Automation*, 2003, 19(5): 765-781
- [2] Bicchi Antonio, Scilingo E P, Rossi D D. Haptic discrimination of softness in teleoperation: the role of the contact area spread rate. *IEEE Trans on Robotics and Automation*, 2000, 16(5): 496-504
- [3] 陈亮亮, 孔祥艳, 程甜甜等. 基于体系结构和面向构件的虚拟仪器开发方法. *仪器仪表学报*, 2007, 28(4): 263-266
- [4] 牛文波, 霍剑青, 王晓蒲. 基于 UI-Model 体系结构的虚拟仪器研究. *中国科学技术大学学报*, 2007, 37(3): 323-326
- [5] 赵江滨, 史铁林, 来五星等. IsVIP: 一种即时可重构虚拟仪器平台. *仪器仪表学报*, 2007, 28(4): 634-638
- [6] 王党校, 张玉茹, 王玉慧等. 集成图形和力觉交互的牙齿探诊模拟系统开发. *系统仿真学报*, 2004, 16(8): 1729-1731
- [7] 陈乐, 杨小虎. MVC 模式在分布式环境下的应用研究. *计算机工程*, 2006, 32(19): 62-64
- [8] 岳永辉, 王莉莉, 郝爱民等. 一种可扩展的图形绘制引擎的体系结构. *系统仿真学报*, 2006, 18 增刊(1): 73-77
- [9] 陈雪峰, 刘松鹏, 张世永. 一种可扩展的分布式虚拟环境体系架构. *系统仿真学报*, 2004, 16(8): 1729-1731
- [10] Erich G, Richard H, Ralph J, et al. 设计模式—可复用面向对象软件的基础. 北京: 机械工业出版社, 2000. 137-143

## An software architecture for haptic interaction based on MVC pattern and plug-in

Liu Jia<sup>\* \*\*</sup>, Song Aiguo<sup>\*\*</sup>

(<sup>\*</sup> School of Information and Control, Nanjing University of Information Science & Technology, Nanjing 210044)

(<sup>\*\*</sup> Remote Measurement and Control Key Laboratory of Jiangsu Province, School of Instrument Science and Engineering, Southeast University, Nanjing 210096)

### Abstract

Based on the analysis of the features of current software for haptic interaction, the paper proposes an extendible software architecture based on the model-view-controller (MVC) pattern and plug-in to solve the critical problems in application. Then three extendible mechanisms, i. e. the model extension based on type encapsulation, the controller extension based on state transition, and the view extension based on publish-subscribe, are given. Developers are able to extend the function of the architecture by such mechanisms easily and quickly, and improve the development efficiency effectively. This architecture is important for standardization of the software for haptic interaction.

**Key words:** haptic interaction, extendible, software architecture, model-view-controller (MVC), plug-in