

20
84-87
A

面向对象的可视化应用平台^{*}

Object-Oriented Visualization Application Platform

王兰波 蔡 勋 李林惠
(国防科技大学计算机系 长沙410073)

科学计算可视化
Visc. 平台

面向对象

TP391.41

摘 要 Visualization in Scientific Computing has been the important developing field of computer graphics. This paper argues the architecture and function of visualization software system using the principles of object-oriented technology, in the direction of system integration, to define a standard structure. So we could use visualization software system as an universal tool in scientific research.

关键词 Visualization, Object-Oriented, Architecture, Message flow

科学计算可视化(Visualization in Scientific Computing, ViSC)是把由数值模拟计算或实验获取的大量数据转化为人类视觉可直接观察的可视化图像,有利于加快数据处理速度,洞察多种物理现象的本质。大多数科学家与工程师因为缺乏编程经验,他们希望拥有高级工具来使他们的精力集中于自己的专业领域。随着可视化研究工作的进展,已出现了一些商用的或学术研究性的系统如 VAS 和 apE,然而科学可视化还处于探索研究的阶段,许多技术细节与标准还有待进一步的完善。本文提出一种面向对象的可视化应用平台的结构与规范,重点突出了其灵活、通用、开放和高可扩充特性,由此而探讨一些可视化系统的标准。我们的目的是使标准的可视化系统成为科学研究的通用工具。

1 可视化与可视化系统

可视化过程一般包括四个阶段:数据过滤、映射、绘制、显示。由于应用领域的广泛性,数据结构与形式有很大差异,这就需要对数据进行过滤预处理,

产生形式化的数据;第二阶段是从形式化的数据集中生成虚拟对象“抽象可视化对象”(AVO),也就是把模拟数据映射到 AVO 的属性上,我们可以把 AVO 看作是在空间和时间上产生扩充的虚构对象,它的属性域包括图元、时间、颜色、透明度、反射度、光照度、体素等等,AVO 的时间和几何域同模拟域中相似量可以不对应;第三阶段是绘制,一般由传统的计算机图形学技术实现画图画绘制,最后是图像以动画的方式显示。

可视化系统的一般结构如图1所示。可视化系统有别于一般的软件系统,也不同于传统的图形系统,这是由于使用对象和使用目的的特殊性。在概念上,我们对此应该有一个新的认识。可视化系统主要包括两方面:一个映射工具集和一个管理操作工具集,它一般应具有这样一些特点:①简单性,具有直接操作功能的用户界面。②完整性。③灵活性,对于不同领域的不同问题都可以在系统中构造一个有针对性的应用程序。④扩充性,能方便地引入新的算法。⑤适应性,能方便地建立在不同计算环境中的平台上。

[3] 蔡希尧、刘西洋等,分布系统与分布对象计算,计算机科学,22(3)1995
[4] 石祥滨、张斌等,基于 CORBA 的异构分布信息系统,小型微型计算机系统,18(2)1997
[5] 蔡希尧、边平定等,The Design and Implementation of Remote Procedure Calls, Proc. of the 3rd Pan Pacific Computer Conf., 1989
[6] OMG, The Common Object Request Broker: Archi-

ecture and Specification, Revision 2. 0. July 1995, Updated July 1996
[7] Microsoft Corp. and DEC, The Component Object Model Specification, Draft Version 0. 9, Oct. 24, 1995
[8] Kraig Brockschmidt, What OLE Is Really About, Microsoft Corp. Web Site, July 1996
[9] Microsoft Corp., Java & COM Integration, Microsoft Corp. Web Site, May 23, 1996

* > 银河巨型机软件开发项目

可视化系统要求建立于一定的环境中,如图2所示。目前各种计算机软、硬件资源日益标准化,已为其提供了良好的外部条件。

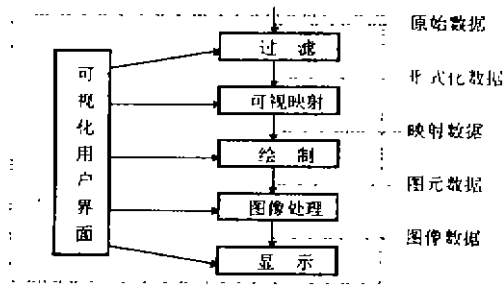


图1 可视化系统模型

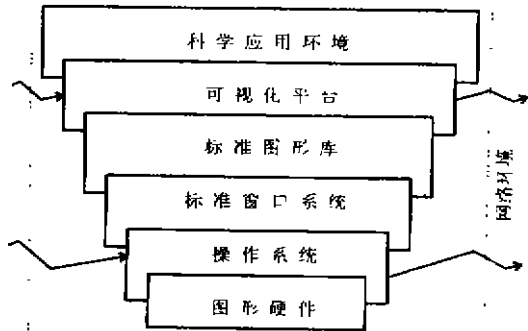


图2 可视化系统环境

为了达到上述要求,可视化系统的设计必须紧紧围绕用户,其处理的中心是用户的数据而不是图形数据,一切与用户无关的细节应隐藏起来,由系统代劳,使用户能交互式地操纵整个系统,提高用户发现问题及解决问题的能力。目前已运行的许多系统为实现这些目的使用了各式各样的体系结构,主要包括:1)数据流网络;2)类 Unix 的管道系统;3)程序性界面系统(Procedural Interfaces)。但这些可视化软件多半是一些低层次的工具软件,大多针对某一特定领域。所以,我们必须探讨可视化的更一般机制以及适应这些机制的更强大的系统结构与方法。从八十年代发展起来的面向对象技术,使我们有可能用以开发出高级、较通用的科学可视化软件工具。

2 面向对象技术与可视化技术

从某种意义上说,可视化是一个认知过程,其认知的根本对象是世界,认知的手段或者说直接对象是数据,认知的直接结果是以一种直观形式表达了客观事物的图像,为研究人员提供了与数据交互的手段。在可视化系统中对象拥有传递和处理信息的能力,并且可以相互作用,从而实现系统的功能。对

象对应于数据单元或处理单元,不仅提供了理解现实、表达现实的一种方法,还提供了系统与用户或其它系统接口的灵活手段。

我们认为面向对象技术与可视化技术存在两个明显的结合点。其一,计算机图形处理已采用了大量的并行算法,面向对象的概念具有天然的并行机制,在程序级上可给予算法实现以强有力的支持,能满足可视化的实时性及处理对象的复杂性所要求的高效图像生成。其二,面向对象的方法学比较自然地模拟了人类认识客观世界的方式,类的各个对象可通过定义一组“方法”来说明该对象的功能,即允许作用于该对象上的各种操作。对象间的联系是通过传递“消息”来完成的。发消息通知对象去完成一个允许作用于该对象的操作,但该对象将如何完成这个操作的细节却被封装在相应的对象类的定义中,对外界是隐藏的。因此,面向对象技术能很好地满足我们对可视化系统的要求。我们可以建造出这样一个面向对象的可视化体系结构,用户只要知道使用所聚合的方法的规则,那么就可以基于自己的专业领域任意地利用所提供的“构件”组合成最适合自己要求的可视化应用实例。

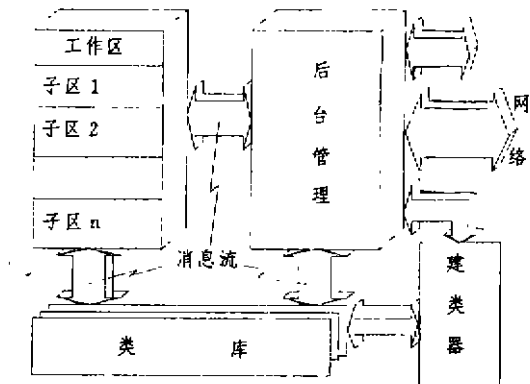


图3 面向对象基于消息流的可视化体系结构

3 系统结构、功能及特点

我们提出的一种面向对象基于消息流的可视化应用平台的体系结构,如图3所示。该系统结构包括四个部分:工作区、类库、后台管理、建类器,前两部分对用户是可见的。在用户搭构可视化应用程序时,用户只需用简单的操作方式(如鼠标)在类库中挑选可视化类送至工作区。在这里,每一个可视化类产生一个或多个具体的实例对象变量,并在后台管理登记,再由用户把它们连接起来。若连接成功用户即可

输入数据,进行整个可视化过程。一个实例对象以一个图符(Icon)形式存在并带有自身的参数控制面板(Panel),对象之间的连接就是消息流管道的定向。这样,多个实例对象之间通过消息流管道的连接就可以构成可视化的四个阶段,当然管道两端数据接口必须匹配。消息流统一了数据流和控制流。当用户输入数据后,依次地各管道内就有消息流在流动。各实例对象受上一级发出的消息流的驱动,对所接受的数据(消息)用已定义的方法进行作用,再形成新消息流向下一级输出。由所有连接的实例对象构成的一个可视化应用实例的执行是动态的。在执行期间,各实例对象通过输入输出缓冲区保存旧的消息流并等待上一级消息流,一旦出现或更新就立刻被驱动,以产生或更新自己的输出消息流。这样,整个应用实例可由最初端口的输入数据激活或由用户改变控制面板上的参数来再一次激活。另一方面,由于每个实例对象是由类库中某个类产生的,它可以产生多个实例对象,所以我们可以把工作区分成若干子区,每个子区的实例对象同属一类,以允许有多个数据源同时进行数据输入,同时进行可视化过程,并行地得到结果,便于用户对同一个问题的不同数学模型进行比较考察。工作区实际是一个面向图符的环境,存放所有的实例对象,并接受用户发出的消息,通知后台管理作相应的处理。用户使用可视化程序设计语言交互式地操纵图符来构造可视化应用程序。

后台管理是系统的核心,为用户完成一切他所不熟悉的技术处理细节,包括语法检查、连接、初始化、激活实例对象、消息传递等等。它还负责整个系统与环境的联系以取得支持。它包括若干个实例对象表、类语法表及一个系统函数聚合,并且这三部分之间及内部框架都是开放的、面向对象的,以方便系统扩充,增强功能。当用户选择可视化类时,后台管理就在送到的那个子区所对应的实例对象表里登记一个名字,当用户进行连接时就在表中加上相应的连接情况,然后调用系统函数集中的语法检查函数,根据已存在的类语法表对连接进行语法检查,即检查输入输出端口是否匹配,并检查整个实例对象集是否已包含了整个可视化过程。实际上,后台管理是检查工作区中存放的若干树结构是否符合语法。当静态检查通过后,一旦用户要求开始可视化过程,后台管理就要遍历整个树结构,调用系统函数初始化每一结点(实例对象)并唤醒适当的结点。另外,这些树结构可以被系统存档以方便下一次使用。后台管

理的一项重要工作是支持实例对象间频繁的消息通讯,其方式有两种:第一是直接调用系统的通讯功能,如 Unix 的管道机制。第二是在内存中为每个对象分配各自的输出消息区。每个区用一个指针指向起点,并建立一个表。当一个实例对象要向下一级传递消息时只需传递一个指针副本。另外每个消息区都有一个计数器,当增加一个访问时就加1,否则减1。当没有实例变量对其访问时,计数器为0,就释放这片内存区。此外,后台管理还要协调、同步各实例对象。若某一实例对象的上一级正在运行还没有发出消息,就让它进入“睡眠”状态,等待上一级完成计算后再将其唤醒,以节省机器开销。从某种意义上说,后台管理充当了类似操作系统的角色。

类库是系统运行的重要基础,提供了可视化应用实例的基本构件。为了使系统能高效、通用,我们必须制定一个类的基本标准,即一个通用的接口标准。它包括两个方面:一是类的实例对象之间进行消息通讯的 I/O 接口标准。二是类的实例对象在工作区里受后台管理的操纵、接受其口令的接口标准。另外,一个类的定义还包括安全公式,给后台管理提供语法规则。当然,类的定义还要有一个方法聚合,对发送过来的消息流进行操作以产生自己要输出的消息流。这里,我们顺便讨论一下消息流模型。消息流包含两部分:数据部分和控制部分。前者应采用系统统一的标准数据格式,而后者则要与下一级的方法集相匹配。系统的类库至少应包含以下几种基本类:第一是输入类,完成数据的输入、过滤并转化成统一格式。由于原始数据格式的多样性,这就要求有多种不同的输入类。第二是映射类,显然也要有多种以便使用不同的算法对不同领域的数据进行处理,即使是同一领域,映射类也可不只一个,因为要支持系统并行性,数据也需并行化。我们可以选择合适的数据模型把输入数据集分割成若干互不覆盖且互相关联的子集。每个集合的数据和它们之间的关系(控制流)构成相互关联的若干消息流,送入相应的映射类实例对象,并行地转化为抽象可视化对象。第三是绘制类,把映射类生成的数据转化成图像。最后是显示类,支持用户对生成的图像进行动画操作等等。另外还可适当地定义一些缓存类,当数据计算量较大,达不到实时动画的速度时,可以把它接在输出口,对数据进行缓存处理。

建类器是对系统进行扩充的主要工具,其作用是把用户引入的新算法的函数进行标准化,通过继承若干基本系统类加入系统的类的标准基本框架,

使其符合通用性要求,并根据新建立类的特点向后台管理提供新的语法规则,另外还要根据用户的要求,为新类设置参数控制面板.如图4所示就是一个标准类的模型.

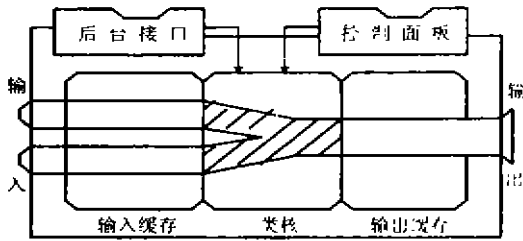


图4 类的标准模型

至此我们已给出了面向对象基于消息流的可视化应用平台的较完整的描述(其形式化定义见附录,略).可以看出该系统环境具有如下特点:①通过分类命名使复杂的协议与重用性变得十分灵活简便;②面向对象的用户界面使系统成为一种简单实用、用户可自定义的软件环境,可按用户要求尽快改变,使编写应用程序与使用两者界限模糊;③一种统一的对象范式,要求系统的每一个单元是一个具有实例变量的对象,并且有状态和作用于某种状态的方法.所有的实例变量和方法都能通过可视化语言交互地访问;④由于统一了对象范式,使得系统具有强大的扩充能力,用户甚至可以根据自己的需要而调整系统的内部结构;⑤统一的对象范式支持网络环境下的并行分布式可视化的实现.

最后,我们还想说明一下后台管理在网络环境下的作用.前面我们基本上描述的是一种集中式的系统.实际上,由于整个可视化体系结构的面向对象的特点包括后台管理的面向对象的框架,这极大地提高了系统的开放性和分布性,我们可以利用网络环境中其它计算资源进行分布式可视化.只要把后台管理或者它的核心加载到网络不同的结点上.并在系统函数集外再加上一个系统协议层,我们就能充分利用 Unix 的网络传输套接字(Socket),其模型如图5所示.例如一个可视化应用的映射类的实例对象需要很大的计算量,可以通过协议请求网络上另一个结点的后台管理为其建立一个实例对象 x,并在本结点的工作区建立一个虚对象 prox,应用程

序中其它实例对象与 prox 的连接管道是逻辑上的,实际管道是通过网络与另一个结点上的实例对象 x 相连,它们之间的同步与协调由两个后台管理共同负责.显然,利用这样的机制也可方便地接受超级计算机通过网络发送来的大体积数据.

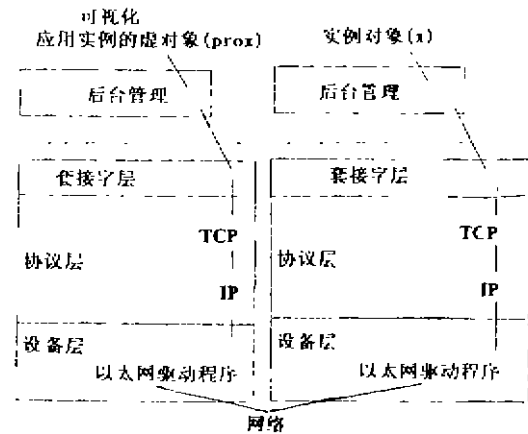


图5 分布式可视化模型

结束语 自 ViSC 报告发表至今,国外在该领域取得了迅速进展.很多超级计算中心、实验室及学术单位纷纷制定了发展 ViSC 技术的长期计划,例如 RIVERS.值得指出的是,在众多已研制开发的 ViSC 软件中,最著名的系统首推 AVS(Application Visualization System)与 apE(Animation Production Environment).apE 和 AVS 都采用了面向对象和数据流设计思想,都是面向应用的可视化系统.

目前,我们正积极开展可视化系统的研究工作,特别是面向对象技术机制的研究.系统的宿主将是高性能图形工作站.我们希望通过制定完整的系统标准与协议及提高网络传输率,在银河高速网上建立一个面向对象的、开放的分布式可视化环境,以提高用户对银河超级计算机所产生的大量数据的处理能力.

参考文献

[1] 汪成为,面向对象分析,设计及应用.国防工业出版社,1992
 [2] Graig Upson, The Application Visualization System, A Computational Environment for Scientific Visualization, IEEE Computer Graphics & Application, 4(6) 1989