

基于 Qt 和 Open Inventor 跨平台虚拟油泥造型系统构建方法的研究与实现^{*}

闫锋欣 侯增选 张定华 孙 蓓

(西北工业大学现代设计与集成制造技术教育部重点实验室 西安 710072)

摘要 采用传统开发流程的应用软件系统一般无法跨平台运行于 Unix/Linux, Mac OS 或 MS Windows 操作系统。在研究软件的跨平台技术基础上, 结合 Qt 应用程序框架和 Open Inventor 图形软件包的特点, 提出一种新的构建跨平台图形应用软件系统的通用技术方案, 并给出了 RedHat Enterprise Linux 4 和 MSWindows 系统下的实现方法。通过描述虚拟油泥造型系统的构建机理, 表明该方法可以满足一般跨平台软件的开发需求。最后着重给出基于 KDevelop 集成开发环境的 VCMS 开发流程, 验证了方法的有效性和意义。

关键词 Qt, Open Inventor, 跨平台, 虚拟油泥造型系统, 通用性开发方法

Research and Implementation on Building Cross-platform Virtual Clay Modeling System Based on Qt and Open Inventor

YAN Feng-xin HOU Zeng-xuan ZHANG Ding-hua SUN Bei

(Key Laboratory of Modern Design & Integrated Manufacturing, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract As a rule, the GUI (Graphical User Interface) applications developed with traditional methods cannot seamlessly run on multi-platforms, such as Unix/Linux, Mac OS or Microsoft Windows. A new method of building cross-platform GUI applications was introduced by studying the cross-platform techniques and using these comprehensive object-oriented C++ frameworks, Qt framework and Open Inventor toolkit. In order to describe their mechanisms in detail, some important implementation processes in RedHat Linux Enterprise 4 and Microsoft Windows XP were also presented. And, the hierarchical model used in VCMS (virtual clay modeling system) indicated that this method could meet the requirements of general cross-platform applications. At last, the details of VCMS development process based KDevelop consequently approved the validity and significance of this method.

Keywords Qt, Open Inventor, Cross-platform, Virtual clay modeling system, Universal development method

随着计算机软件产业的飞速发展,人们已经开发了大量的应用软件,但存在总量丰富而重用率低下之间的矛盾。因此,为了能够有效地满足面向多平台的复杂应用对不同用户多样服务的需求,将不同系统平台下广域分布的异构、自治资源进行按需组织和管理,以实现动态、协同地提高资源综合利用率和资源共享率,这已成为一个重要的科学问题^[1]。

近年来, Linux 以其自由开放性和稳定性逐渐获得了程序开发人员的青睐,但其单调的命令执行程序开发环境极大程度上阻碍了大型程序的规模化发展。与之相对应的是,近年来基于 Windows 平台的开发环境则经历了跨越式的发展,完成了从单语言、单环境到交叉开发、系统集成的过渡,用户的代码量呈现几何级数增加。因此,在充分利用 Linux 程序源码开放和自由使用的基础上,增加其开发环境的易用性和应用软件的跨平台特性将可能有效解决上述问题。

Qt 应用程序框架^[2]和 Open Inventor 工具包^[3]均具有良好的跨平台特性,在源代码级可以做到“一次编码,随处编译”的跨平台效果。从跨平台技术角度讲, Qt 框架为用户提供了诸如应用进程管理、对话框、多线程等的全面解决方案,如刘汇丹等^[4]利用 Qt 开发了可以显示蒙、维等多种少数民族文

字的国际化图形用户界面(GUI); R. Brun^[5]等基于 Qt 完成了跨平台的交互式应用软件系统。Open Inventor 是从 OpenGL 派生出来的可独立于系统平台的软件包,既可有效利用系统的硬件资源,又能简化程序开发的代码量,其扩展库(如 SoWin 等)为用户提供了便捷的 GUI API,大大提高了三维场景的构建效率和管理效果。

本文将通过研究分析跨平台技术,结合 Linux, Qt 和 Open Inventor 程序设计特点,采用 VC 和 KDevelop C/C++ 集成开发环境(IDE),从跨平台应用程序开发的角度,讨论基于跨平台技术有效构建三维图形应用系统的关键技术。第 1 节概述跨平台技术;第 2 节分析软件跨平台所需的主要技术,结合虚拟油泥造型系统 VCMS(Virtual Clay Modeling System)^[6],给出其跨平台实现的技术途径;第 3 节给出 VCMS 的构建过程和程序实现流程;第 4 节给出 VCMS 在不同平台的运行效果;最后是分析结论。

1 跨平台技术

对于一个应用程序,所谓的跨平台是指其能够独立运行于不同的操作系统平台,如 Windows, Unix/Linux 以及 Mac

^{*} 本文研究得到国家“863”计划项目(2006AA04Z111),国家自然科学基金项目(50475145, 50775185),西安市科技计划项目(GG06027)资助。闫锋欣 助教,博士研究生,主要研究方向为计算机辅助技术、虚拟产品开发;侯增选 副教授,德国柏林工业大学博士,主要研究方向为虚拟现实技术、五坐标数控加工仿真等;张定华 博士,教授,博士生导师,主要研究方向为快速反应制造、工业 CT 与检测、体视化技术等;孙 蓓 硕士研究生,主要研究方向为虚拟交互技术。

OS,从而有效提高软件的重用性。但由于应用程序在运行中需要综合利用空间、图形、数据库等多项技术,且 CPU 调用指令的解释在不同平台之间也存在差异,所以很难获得完全能够独立运行于不同系统平台的应用程序。但是,我们可以从技术层面获得相应的跨平台技术实现。

1.1 操作系统级的跨平台技术

分层设计的思想可以在操作系统级上实现跨平台。事实上,分层设计思想是网络和计算机系统的一个主要特点,这种方法使层与层之间相互独立,并按照特定的接口进行通信^[7]。例如,CPU 指令集构成了操作系统和硬件之间的接口,系统调用则构成了应用程序和操作系统之间的接口。虚拟机技术是操作系统级跨平台技术的典型代表之一,它通过新增的虚拟中间层截获上层软件对底层接口的调用,并且对该调用重新做出新的解释和处理,以实现异构环境中资源的可共享、可管理和可协同,做到“一次编译,到处运行”,实现目标代码级和源代码级的平台无关性。

1.2 GUI 的跨平台技术

GUI 无法直接跨平台,是应用系统无法实现跨平台的主要原因之一,因为各种操作系统是通过不同的 API 调用函数来生成应用程序各自图形界面的。例如,Windows 平台中运行的应用程序,其 GUI 通常由 MFC 或者 Windows API 构建而成,显然无法运行在 Unix/Linux 下以 X11 库为基础的 X-Window 系统;反之亦然。因此,要使 GUI 具有跨平台能力,则必须通过具有跨平台能力的 C/C++ 等高级语言,利用它们的平台无关性来模拟不同平台中 GUI 的界面模式,以构建具有跨平台能力的 GUI。这也是 Qt 应用程序框架等跨平台软件包实现跨平台的主要机制之一。

1.3 其他跨平台技术

组件技术、中间件(Middleware)技术等也是常用的跨平台实现技术^[8]。对于组件技术,常用的组件规范有 COM, CORBA 和 JavaBean,其实现规范主要由微软,OMG 和 SUN 公司完成,分别用以完成特定功能的、符合二进制标准的软件模块。中间件是位于操作系统和应用程序之间的软件,不同中间件的通信则由通信中间件来完成,从而形成跨平台间的通信,如数据库访问中间件、网络中间件等。

2 跨平台软件实现的技术途径

由于移植性、稳定性和硬件兼容性是衡量跨平台软件实现技术的重要指标,而硬件平台的升级、维护等状态则影响着软件的持续稳定运行,所以屏蔽硬件资源的动态变化对软件的影响就成为保证软件运行稳定可靠的重要问题。下面以 VCMS 为例,对软件的跨平台实现过程做简单讨论。

2.1 VCMS 概述

VCMS 是西北工业大学和柏林工业大学联合研制的汽车车型创新设计系统。该系统基于压缩体素模型表示汽车的虚拟油泥模型,在计算机生成的虚拟环境中,设计人员利用虚拟油泥造型工具(虚拟的刮刀、刮片和模板等),通过对虚拟油泥模型进行刮削、填补、刮补等操作完成汽车外形建模,克服了传统车身设计软件对设计师创造能力的限制,为解决计算机辅助工业设计中的问题提供了有效途径。

2.2 解决 VCMS 跨平台的主要问题

在计算机内部采用压缩体素模型的 VCMS 对于系统稳定性有着苛刻的要求,因而在开发中,硬件平台采用高性能的 SGI 图形工作站;系统主体开发语言采用 ANSI C/C++,并基于 X-Window 的 X11 库实现硬件系统的底层工作;图形用户界面采用 ViewKit 语言^[9],并通过 VkApp 等应

用程序类管理系统的消息响应;在三维图形的显示和操作管理方面,则通过便捷的 Open Inventor API 工具包来完成。

从系统级来看,VCMS 采用的 ViewKit 和 Open Inventor,都是由许多面向对象的 C++ 类构成的软件工具包,而且后者是基于 OpenGL 开发而成的 API,具有天生的跨平台特性,因而由它们完成的系统图形显示和人机交互功能在源代码级都具有跨平台能力。因此,需要尽量减少对基于 X11 库完成系统底层交互功能的代码量。VCMS 开发初期,由于系统开发体系采用人工管理方式,造成系统原型缺乏版本控制能力,因而适当的集成开发环境也是跨平台开发中需要考虑重要因素之一。

2.3 VCMS 跨平台技术的选择

针对 VCMS 自身的特点和对跨平台技术的要求,以“多平台源码一致”为原则,对各层次跨平台技术的选择如下。

(1)针对操作系统:不采用依赖运行环境的 Java,选择 C/C++,实现多平台共用同一套代码,做到“一次编写,多次编译运行”。其优点是性能可靠,与不同平台的集成性高,便于系统控制。

(2)针对图形相关模块:图形的显示和控制选用基于 OpenGL 的 Open Inventor 图形软件包;GUI 部分选择跨平台的 Qt 应用程序框架,并负责 VCMS 系统的程序管理、事件响应等。

(3)针对集成开发环境:Windows 下选择 VC,便于实现系统管理和协同开发;Unix/Linux 下选择 KDevelop。两者对于源代码文件的管理方式相似,易于实现代码同步。

2.4 图形用户界面:Qt

Qt 使用“一次编写,随处编译”的方式为开发 GUI 的用户提供了一个完整的 C++ 应用程序框架,并以其卓越的兼容性、高效的 C++ 性能和丰富的 API 而成为 Unix/Linux 平台中 GUI 开发的事实标准。程序开发人员使用 Qt 的单一源程序可以构建出能够独立运行于从 Windows 98 到 XP,从 Mac OS X, Solaris, HP-UX 到 Linux 以及其它基于 X11 库的 Unix/Linux 上的应用程序。Qt 的双重授权机制^[2]及其丰富的产品线,为各行业的开发人员提供了极大便利:非商业用户可以基于 QPL 和 GPL 开发自由软件,商业用户则可以通过购买的 Qt 商业许可证开发商用软件。

Qt 设计器(Qt Designer)可用于开发更为复杂的 GUI 窗体。Qt 设计器 4.3.2 在 Windows 系统中的运行界面如图 1 所示:用户可以方便地从左侧的工具箱中选择适当的控件,将其放置在窗口中的大致位置,然后通过设计器中提供的 QLayout 等完善对齐方式完成界面的最终布局。设计器能够大大减少手工编程的代码量,显著提高程序开发的效率。

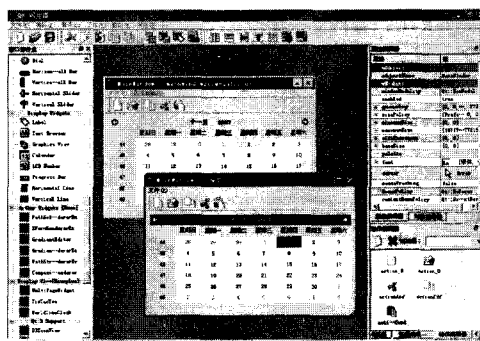


图 1 运行于 Windows 系统的 Qt 设计器

2.5 三维场景的显示和管理:Open Inventor

Open Inventor 是目前世界上应用最为广泛的面向对象

和交互式的三维图形软件开发包。它基于 OpenGL 图形库开发而成,具有较好的硬件兼容性和较强的跨平台能力。SGI 在其 UNIX 操作系统下对 Open Inventor 进行了原型开发, Mercury 和 SIM 公司则将其扩展到 Windows 和其他 Unix/Linux 平台。因此,Open Inventor 既有 OpenGL 的平台无关性,又为用户提供了功能更为强大、代码更为简练的图形 API。图 2 给出了目前用于 Windows 和 Linux 的 Open Inventor 组成结构。图 3 给出的是 Mercury 公司 Open Inventor 的扩展模块。

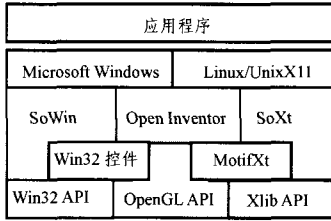


图 2 Open Inventor for Win/Xt 的组成结构

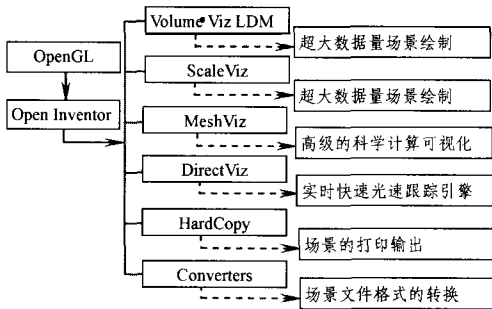


图 3 Mercury Open Inventor 的扩展模块

2.6 集成开发环境:KDevelop

功能完善的跨平台集成开发环境,目前尚没有理想的选择。在 Windows 系统中,一般选择 VC 或 C++ Builder 作为标准 IDE,它们均可以提供强大的项目管理功能。在 Unix/Linux 系统中,可用的 IDE 有 KDevelop^[10], QDevelop^[11] 和 SlickEdit^[12]。KDevelop 可以提供类似于 VC 的开发环境,功能相对比较完善,是常用的标准 IDE;QDevelop 是一款完全面向 Qt 4 的开发工具,其功能目前还很不完善,暂且无法真正应用于实际的大型项目开发过程;SlickEdit 是一款跨平台的 C# 开发工具,但并不提供对 Qt 的直接支持,而且在软件开发时都需设置相应的配置文件。因此,一般在 Unix/Linux 中选用 KDevelop 作为标准 IDE。

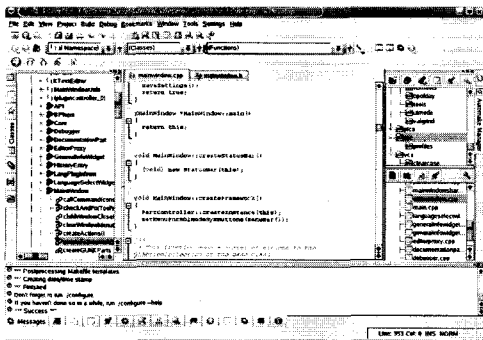


图 4 KDevelop 3.4.1 在 RedHat Linux 系统中的开发效果

KDevelop 是一款功能强大的开发工具。图 4 给出了 KDevelop 3.4.1 在 RedHat Linux 系统中的运行示意图。在 KDevelop 中,开发人员可以利用左侧的类视图、文件列表等

观察项目的组织结构和源代码的存放形式,在工作区中部编辑打开文件的源代码;右侧给出了 QMake 管理器等工具,可以完成工程联编时所需的配置工作;下侧是信息观察区,可以观察工程的编译、连接情况,也可以在终端模拟器中直接执行终端命令。此外,基于 Doxygen 文档生成系统,KDevelop 可以为工程输出包括 HTML, latex, RTF, ps 以及 manpage 等格式的项目文档。

3 跨平台 VCMS 的实现流程

3.1 Qt 与 Open Inventor 的结合

设计 Open Inventor 的初衷之一就是要兼具 OpenGL 的可移植性和窗口系统无关性。如同 OpenGL 自身使用了大量的与平台无关的类和少量的与窗口系统相关的类那样,Open Inventor 也提供了少部分与窗口系统相关的类:SoXt 类适用于 Unix/Linux 环境,SoWin 适用于 Windows 环境,而 SoQt 则针对 Qt 环境。由于 SoQt 类是 Open Inventor 图形渲染核心库和 Qt 库的结合,所以由其构建的源代码可以无缝运行于多个操作系统平台。图 5 给出了两者的结合示意图。

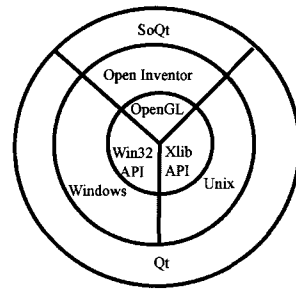


图 5 Qt 和 Open Inventor 的结合

在应用程序中,使用 Qt 和 Open Inventor 的步骤可以表示如下:

```
void main(int argc, char * argv[]) {
    // 使用 SoQt::init()初始化 Open Inventor
    QWidget myWindow = SoQt::init(argv[0]);
    // 创建场景的根节点,加载场景图
    SoSeparator theRoot = new SoSeparator;
    theRoot->ref();
    // 添加其他节点(sepTool, sepClay 等)
    theRoot->addChild(sepTool); // 刀具子场景
    theRoot->addChild(sepClay); // 油泥模型
    // 刀具和油泥模型做布尔运算构造造型结果
    theRoot->addChild(volTool + volClay);
    // 创建一个 Open Inventor 渲染区或者观察器
    SoQtExaminerViewer * myViewer = new
    SoQtExaminerViewer(&myWindow, "viewer");
    myViewer->setSceneGraph(theRoot);
    // 使用 show()显示窗口
    myViewer->show();
    // 进入 Qt 的事件循环
    SoQt::show(myWindow);
    SoQt::mainLoop();
}
```

3.2 创建 VCMS 的场景图

在 Open Inventor 中,所要显示的一个或多个场景存储在场景数据库(scene database)中,可以通过场景图(scene graph)来表达其结构形式。节点(node)是构成场景的基本单元,可以代表场景中的一个几何对象、属性对象或其组合对象。也就是说,场景图使用节点来表达场景中的所有物体以

及物体之间的相互关系。在 VCMS 中,需要在三维场景中生成模拟的工作环境,如辅助网格、虚拟刀具和油泥模型等节点。由于 VCMS 中的工具和油泥模型需要根据用户的需要而能够任意更换或者改变,所以可把它们作为场景图中的子场景来加以处理,如图 6 所示。

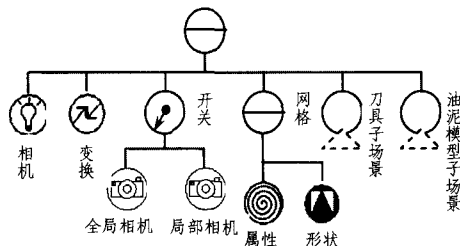


图 6 VCMS 的场景简图

3.3 造型中视点的控制

考虑到虚拟油泥造型过程中需要不断刮削或者添补油泥模型,并且设计人员会根据其自身感性认识不断修改模型的造型效果,都要求给用户提供一种方便易操作的交互手段。另一方面,出于完全模拟真实油泥造型效果等特殊目的,也可能需要对整个场景进行位置角度的精确控制。因此,造型过程的便捷性和场景视点的控制易用性是虚拟油泥造型系统的重要研究内容。

在硬件设备上,我们选用三维鼠标^[13]来控制虚拟场景中视点的平移和旋转。三维鼠标能够监测到手指施加其上的很轻微的力,并将其分解成 X, Y, Z 方向上的平移量和转动量;通过获取这些量来构成场景的平移矩阵 T_{xyz} 和旋转矩阵 M_R 。用户通过变换矩阵 M 与 T_{xyz} 和 M_R 的积求得场景的最终变换结果:

$$\text{平移: } P_{new} = (P_{old}) [T_{xyz}] \quad (1)$$

$$\text{旋转: } P_{new} = (P_{old}) [M_R] \quad (2)$$

其中, T_{xyz} 和 M_R 均为采用齐次坐标形式的 4×4 变换矩阵。此时,用户则利用空间球的旋转、移动等直观操作方式,大大提升了空间操作的效率。

因此,使用三维鼠标能够方便地对(子)场景进行旋转、移动等操作,可以提供良好的 6 自由度交互控制方式。

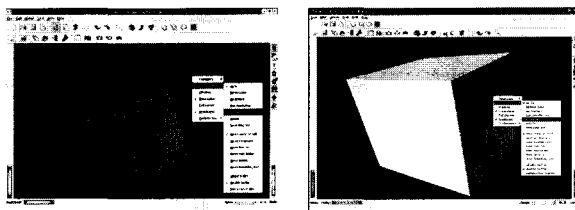
3.4 系统的程序实现

为确保良好的源代码级跨平台特性,VCMS 开发中严格实施 ANSI C++ 标准和基于面向对象的编程规范。系统的整体结构遵循 Qt 框架标准,利用大量的自定义类和 C/C++ 的继承、派生完成系统底层功能,利用 Qt 中的类和方法完成系统所需的用户交互、事件管理等功能,最后利用 Open Inventor 实现结果的显示和控制,因而可以确保源代码的移植性和硬件兼容能力。

4 运行效果比较

图 7 给出了 VCMS 在不同平台下的运行效果。图 7(a) 是 VCMS 在 Windows 下的运行效果图,GUI 采用 Qt 4.3.2(商业版),图形核心采用 Open Inventor 5.0,分辨率为 1024×768 ,开发环境为 VC 6.0;图 7(b) 给出的是 VCMS 在 RedHat Linux 系统下的运行效果,GUI 采用 Qt 4.3.2(open source 版),图形核心采用 Open Inventor 6.1.2,分辨率为 1280×1024 ,开发环境为 KDevelop。在运行的 VCMS 系统中,通过鼠标拾取场景中的油泥模型或刀具,用户利用三维鼠标可以方便地对其进行平移、旋转;当鼠标点取场景(不拾取物体)时,则可以对场景整体进行旋转操作;在达到操作视角时,用户则可以利用普

通鼠标完成油泥模型的刮削、填涂和刮补操作。



(a) Linux 系统

(b) Windows 系统

图 7. VCMS 在不同平台下的初始运行效果

结束语 为了解决应用程序跨平台运行的问题,本文提出了一种基于 Qt 和 Open Inventor 并在 VC 和 KDevelop 集成开发环境中构建了可跨平台运行的 VCMS,也适用于其他三维图形应用系统的开发过程。该方法利用 Qt 和 Open Inventor 完善的跨平台机制,构建了可跨平台运行于 Windows 和 Linux 操作系统的 VCMS 系统。该方法利用 Linux 系统下自行编译的 KDevelop 3.4.1 集成开发环境,较好地解决了利用 GCC 和 Makefile 开发程序时缺乏 IDE 的问题,为 Linux 下程序开发提供了新方法。

限于实验室现有的软硬件平台和对 Mac OS 下开发环境的掌握,我们目前尚无法完成 Mac OS 平台下 VCMS 的兼容性验证。同时,由于我们现有的 Windows 平台下的 Open Inventor 6.0 与 Qt 的开发模块只能二进制兼容(binary compatible)于 Qt 3.1.2,使得 Linux 和 Windows 系统下的 API 调用并不完全相同,但随着 TGS 公司对 Open Inventor 的研发,我们相信在不久的将来一定可以顺利解决这一问题。最后,由于本方法是基于源代码级的跨平台方案,所以应该可以顺利实现 Mac OS 的跨平台应用。

参考文献

- [1] 怀进鹏,李沁,胡春明. 基于虚拟机的虚拟计算环境研究与设计[J]. 软件学报,2007,18(8):2016-2026
- [2] Blanchette J, Summerfield M. C++ GUI Programming with Qt4[M]. [S.l.]: Prentice Hall PTR,2006
- [3] 闫锋欣,侯增选,张定华,等. Open Inventor 程序设计[M]. 北京:清华大学出版社,2007
- [4] 刘汇丹,芮建武,姚延栋,等. 基于 Qt 的国际化图形用户界面设计与实现[J]. 中文信息学报,2006,20(4):94-99
- [5] Brun R, Fine V, Lauret J, et al. Cross-platform approach to create the interactive applications based on ROOT and Qt GUI libraries[J]. Nuclear Instruments and Methods in Physics Research Section A,2004,534(1/2):94-97
- [6] 张淑焕,侯增选,孙根正,等. 虚拟油泥造型方法[J]. 计算机应用研究,2007,24(2):181-182
- [7] Smith J E, Nair R. The architecture of virtual machines. IEEE Computer,2005,38(5):32-38
- [8] 赵斌,李欣,朱美正. GIS 跨平台技术研究与实现[J]. 计算机工程与应用,2005,41(23):193-195,212
- [9] Integrated Computer Solutions, Inc. The ViewKit Programmer's Guide[EB/OL]. (2002-04) [2007-11-15]. www.ics.com/support/docs/viewkit/2/viewkit2prog.pdf
- [10] http://www.kdevelop.org/ [OL] [2007-10-12]
- [11] http://qdevelop.org/ [OL] [2007-10-16]
- [12] http://www.slickedit.com/ [OL] [2007-11-17]
- [13] 3Dconnexion Inc. 3D Motion Controller UNIX SDK Guide (V3.2) [EB/OL]. (2002-04) [2007-11-03]. http://www.3dconnexion.com/docs/13dunix_sdk.pdf