文章编号:1005－0523(2006)04－0099－04

Pro/Toolkit 的二次开发技术研究及应用

周慧兰,周新建

(华东交通大学 机电工程学院,江西 南昌 330013)

摘要:Pro/E 中的 Pro/Toolkit 工具包是用户应用程序开发者的 API,具强大的二次开发功能,它可为用户定制优良的用户界面,本文详细介绍了应用 Pro/Toolkit 工具进行二次开发的关键技术和方法.

关 键 词:Pro/Toolkit;二次开发

中图分类号:TP391.72 **文献标识码**:A

0 前言

二次开发的目的是设计一个软件系统或称之为设计工具来辅助具体的设计.常用二次开发支撑软件及其二次开发工具有 PRO/E 的 Pro/Toolkit(简称 Pro/T)、UG 的 Open API、Solidworks 的 Solidworks API 等.其中,PRO/E 软件是非常成熟的 CAD/CAM 商用软件,功能全面,性能优良,更重要的是它提供的 Pro/Toolkit 工具开发语言为通用的高级语言,编译环境优良,常作为二次开发的支撑软件.

1 Pro/Toolkit 简介

Pro/T 提供了一个很大的 C 函数库,利用其库函数编写的外部应用程序可方便又安全地访问 Pro/E 的数据库及其应用程序,进行二次开发,扩展其功能.因此,也可以把 Pro/T 看成是 PTC 用户应用程序的界面(API).

2 Pro/Toolkit 应用程序模式

Pro/T 应用程序主要有两种模式:同步模式(synchronous modes)和异步模式(Asynchronous mode).两者的区别如表 1.

表 1 同步模式和异步模式的比较

比较项目	同步模式	异步模式
通讯方式	发送消息请求操作	远程调用
控制权	交替控制,当一个进程执行操作时,另一进程处于等待状态	PRO/E 与 PRO/T 应用程序两进程之间并行工作
应用程序的启动	PRO/E 根据注册文件启动应用程序,可以在辅助应用程序的对话框中启动	应用程序可以独立于 PRO/E 启动,启动后再启动 PRO/E 或连接到 PRO/E 的一个进程中.不会出现在辅助应用程序的对话框中
运行速度	消息映射机制简单,但运行速度较快消息映射机制复杂,且由于是远程调用,所以速度较慢	消息映射机制复杂,且由于是远程调用,所以速度较慢.
运行方式	以 PRO/E 系统为主,应用程序作为其子模块运行	以应用程序为主,由应用程序根据要求调用 PRO/E 系统

由于在本论文中应用的主要是同步模式,所以在以后介绍的主要是和同步模式相关的技术.

同步模式又可分为两种:动态连接库模式(DLL Mode)和多过程模式(Multiprocess Mode).

1) 动态连接库模式(DLL mode)

Pro/T 应用程序代码集成到 Pro/E 中所用的标准方法是通过应用动态连接库(Dynamically Linked Libraries),当编译 Pro/T 应用程序并和 Pro/T 库相连接时,便创建了一个目标库文件,当 Pro/E 启动时,该库文件便连接到 Pro/E 的可执行文件,这种模式称为“动态连接库模式(DLL mode)”,其运

收稿日期:2006－04－20

作者简介:周慧兰(1974－),女,江西高安人,讲师, 研究方向:CAD/CAM 及二次开发.

行原理如图 1.

在 DLL 模式下,Pro/T 应用程序与 Pro/E 之间的数据交换是通过直接的函数调用进行的.

2) 多过程模式(Multiprocess Mode 或 Spawned Mode)

Pro/T 也支持集成的第二种方法:多过程模式(Multiprocess Mode),也称分离模式(Spawned Mode).在多过程模式下,Pro/T 应用程序代码被编译和链接生成一个独立的可执行文件.这个可执行文件被设计为由 Pro/E 产生且作为 Pro/E 会话期的一个子过程运行,其运行原理如图 2.

在多过程模式下,Pro/T 应用程序与 Pro/E 之间的数据交换是通过进程信息系统(interproecss messaging system)完成的,即通过传递一些必要的信息来激发函数的直接调用,而这些必要的信息、就是比较两个进程之间的函数和它的参数值.

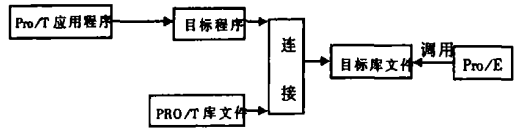

图 1 动态连接模式

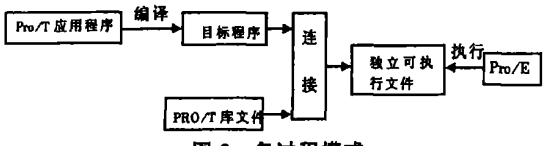

图 2 多过程模式

表 2 动态连接库模式和多过程模式的比较

各比较项	DLL 模式	Spawned 模式
Pro/T 应用程序编译和连接后生成的文件	DLL 动态链接库文件	独立的可执行文件即 EXE 文件
应用程序的启动	在 Pro/E 启动时启动,和 Pro/E 连接在一起	可以和 Pro/E 一起启动,也可在 Pro/E 运行过程中根据用户的需要启动,此时是作为 Pro/E 的一个子程序运行.
Pro/T 应用程序与 Pro/E 之间的信息交换	通过直接的函数调用	通过进程间的通讯机制
Pro/T 应用程序的调试	需将 Pro/T 应用程序和 Pro/E 载入调试器	只需将应用程序载入调试器,而不需要将 Pro/E 载入
内存开销	更小	更大,尤其是当 Pro/T 应用程序频繁调用 Pro/T 库函数时
使用	用户使用阶段,运行速度更快	开发和调试阶段使用

与 DLL 模式相比,多过程模式需要更多的通信开销,尤其是当 Pro/T 应用程序频繁调用 Pro/T 库函数时.但是,多过程模式可以提供以下优点:可通过一个源程序代码调试器来运行 Pro/T 应用程序,而无须将整个 Pro/E 可执行文件装入调试器中.

对于动态连接库模式(DLL Mode)和多过程模式(Multiprocess Mode)的比较如表 2.

3 Pro/Toolkit 应用程序结构

1) 应用程序的头文件

头文件 ProToolkit.h 中定义了 wchar __t,这是宽字符串的数据类型,并为许多其它的头文件和应用程序所引用,所以在每个 Pro/T 应用程序的源代码文件中必须包含头文件 ProToolkit.h,且它必须放在所有头文件的前面.

此外,如果在应用程序中用了某些函数,就应包含这些函数原型的头文件.否则在编译时就不会对函数的参数类型做检验.另外,如果在 Pro/T 应用程序中用到 Pro/Develop 的函数,就需要在 ProToolkit.h 之前包含 Prodevelop.h.

2) Pro/T 应用程序的主程序

Pro/T 应用程序的主程序与一般 C 语言的程序有所区别,它无须包括 main 函数,而必须包含两个称为 user_initialize()和 user_terminate()的函数,分别在 Pro/T 开始和 Pro/E 会话结束时被调用.

函数 user_initialize()的调用,是在 Pro/T 应用程序的初始化和图形窗口的生成之后,在用户与 Pro/E 进行交互的菜单显示之前.而函数 user_terminate()是在 Pro/E 会话期结束时被调用的,它的返回类型为 void.

3) 用户添加的函数体

这是用户自己根据需要添加的函数代码.

4 Pro/Toolkit 应用程序编译、连接和注册

4.1 应用程序的编译和连接

编译和连接 Pro/T 应用程序所需的 C 编译器选项和系统库在每个平台上都是不同的.源程序中为了保证 Pro/T 应用程序所用的 make 文件使用正确的选项,Pro/T 提供了一个安装测试的应用程序,其源程序代码在<Protoolkit>/protk __appls/pt __install __test 目录下,相应的 make 文件是<Protoolkit>/<MACHINE>/obj/make __install.其中<Protoolkit>是 Pro/T 的安装目录.当需要 make 文件时,可将此文件拷贝到包含自己 Pro/T 应用程序源代码文件的目录下,然后对相关选项根据要求进行编辑即可.

其中,make 文件是用来指定源文件如何进行编译和连接,并最终生成可执行文件(*.exe)或 DLL 文件(*.dll).

本系统所用的是 Windows 平台,采用 VC6.0++做调试器.

采用 VC6.0++作为 Pro/T 调试器有两种方法,一种是根据 make 文件直接编译和调试程序;另一种则不需要编写 make 文件,直接由 VC6.0++建立 Pro/T 应用程序项目,并进行编译和连接等工作.实际上两种方法的实质和功能是一样的.

下面对这两种方法进行介绍.

方法一 直接采用 make 文件进行编译工作,需要编写好源文件(包括 C 程序源文件和资源文件)和 make 文件,步骤

如下：

1) 将 make 文件改名为 *.mak 文件，用 VC 打开此文件并建立相应的工程项目.

2) 执行 VC 主菜单命令 Build/Build All，编译连接生成需要的可执行文件或 DLL 文件.

方法二 直接由 VC 建立并编译 Pro/T 应用程序项目，步骤如下：

1) 编写 C 语言源程序，打开此程序，运行选择 VC 主菜单命令 Build/Build 生成一个默认的工程项目.

2) 选择 VC 主菜单命令 Tools/Options，系统弹出 Options 对话框，打开 Directories 选项卡，在 Show directories for 下拉列表框中分别选择 Includes files 和 Library files，并分别添加相应的包含文件和库文件路径.

3) 择 VC 主菜单命令 Project/Settings，系统弹出 Project Settings 对话框.打开 Link 选项卡，在 Category 一栏选 General，在 Output file name 编辑框中填写输出文件名称，在 Object/library modules 文本框中加入对应的库文件.将 Category 切换至 Customize，选中 Force files out 设置强制输出.

4) 其中第二种方法利用 VC6.0++集成环境将 make 文件的编译和连接隐含在开发环境中.

4.2　应用程序的注册和运行

编译连接成功生成可执行程序后，需要先进行 PRO/T 应用程序的注册，然后才能运行.

注册 Pro/T 应用程序，就是向 Pro/E 系统提供该程序的相关信息，即告诉 Pro/E 此应用程序的可执行文件、菜单资源文件和对话框资源及信息资源文件在哪里、以及此程序所依据的 Pro/T 的版本信息等.为此，需制定一个注册文件，通过该文件实现应用程序的注册.注册方式有两种，一种为自动注册，另一种为自动注册.

1) 自动注册和运行

自动注册分两种情况：一是必须将注册文件名取为 protk.dat，并保存于 Pro/E 安装目录的 \ text 目录，或者位于 Pro/E 的起始位置设定的目录.二是在 config.pro 文件设定注册文件(系统变量名为 toolkit _registry _file).

2) 手动注册和运行

选择 Pro/E 界面上的 Utilities/Auxility Applications 选项，选择“Register”按钮注册应用程序.注册成功后选择“start”按钮运行应用程序.

与自动注册方式相比，手动注册方式可以在不关闭 Pro/E 系统的前提下反复修改应用程序.

5　Pro/Engineer 与 MFC 的接口开发技术

从上面的 Pro/E 二次开发的过程上来看，若采用 C 语言来开发，在 Visual C++6.0 环境下编译，人机交互界面设计相对烦琐.此外，一个好的 CAD 系统要和其它系统集成往往需要通用数据库接口，但是至今为止，Pro/Develop 及 Pro/Toolkit 仍没有提供数据库编程接口.为此，这里研究并开发了 Pro/T 与 MFC 的接口，可利用 MFC 强大的功能实现对话框的开发与数据库的访问.

1) 基本 DLL 理论

一般而言，DLL 是磁盘上的文件，它由全局数据、编译的函数及资源组成，它们为进程的一部分.DLL 编译后，加载到首选的地址上.如果与其他 DLL 之间没有冲突，则该文件映射到进程的同一虚拟地址上.DLL 有各种导出函数，客户程序导出这些函数.Windows 在加载 DLL 时对导入与导出进行匹配.

在 DLL 代码中，必须象下面这样声明导出函数：

__declspec(dllexport) int UserFunction();

在用户方面，要求象下面这样声明相应的导入函数：

__declspec(dllimport) int UserFunction();

其中函数 UserFunction 是用户自己定义的函数.

2) Pro/E 与 MFC 接口开发

从本质上来说，Pro/E 与 MFC 的接口就是 Pro/E 系统调用 MFC 应用程序的途径，这里采用动态链接库(DLL)方式可以实现三者之间的通信，因为通信是通过直接的函数调用实现的，所以有执行快的特点，接口方案如图 3 所示.

图 3　Pro/T 二次开发应用程序调用方式

3) 具体实现方法

(1) 使用 CwinApp 类来生成第一个 DLL 工程，名为 PTC.DLL，工程选用共享 MFC 常规 DLL(Regular dll using Shared MFC DLL)选项，然后在此工程中加入 Pro/T 程序，主要是 user _initialize()函数代码.

(2) 使用同样的方法与选项生成第二个 DLL 工程，名为 MFC.DLL，并在此工程中用类向导(ClassWizard)和资源编辑器(Resource Edit)生成所需的 MFC 类(如对话框类，数据库类等)，并定义一个函数完成该类的初始化.

(3) PTC.DLL 中的 Pro/T 程序中调用 MFC.DLL 的导出函数，这是接口实现的关键.加入 Pro/T 程序所用 到的库，如 protk.dll，protoolkit.dll，mpr.lib，… debug mfc.lib 等并指出其路径且设为强制输出，使用 MFC 的编译选项，对两个工程进行编译，生成新的 PTC.DLL 和 MFC.DLL.

(4) 在 Pro/E 中，用 DLL 方式加载 Pro/T 程序 PTC.DLL，再通过 Pro/T 程序调用 MFC 应用程序 MFC.DLL.

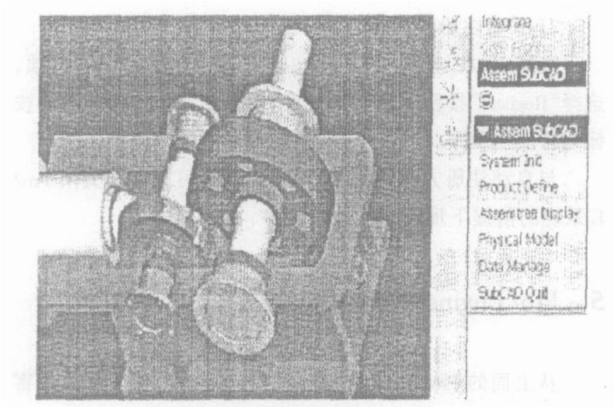

图 4 装配模式下的用户菜单 Assem SubCAD

6 应用实例

本实例应用 PRO/E 二次开发工具开发了装配模式下的用户菜单 Assem SubCAD，该菜单下又分别建立了五个子菜单，如图 4，每个子菜单分别根据用户的需要实现不同的功能.

下面简单介绍该实例的应用过程.

1) 在 VC6.0++环境中应用 C(C++)编辑用户应用程序原代码，部分代码下：

图 4 装配模式下的用户菜单 Assem SubCAD

```
int user __initialize(argc, argv, version, build, errbuf)
int argc;
char *argv[], *version, *build;
wchar __t errbuf[80];
{int menu __id;
ProMenuFileRegister("assembly","assembly.mnu",&menu __id);
ProMenuAuxfileRegister("assembly","assembly.aux",&menu __id);
ProMenubuttonActionSet("assembly","AssemSubCAD",
(ProMenubuttonAction)AssemSystem,NULL,0);
return (0);
}
void user __terminate()
{printf("user terminate!  n");}
```

代码中的 assembly.mnu、assembly.aux 分别对应菜单文本文件.

2) 采用 VC6.0++作为 Pro/T 调试器根据 make 文件直接编译和调试用户程序，make 文件的部分内容如下：

```
# Executable names
EXE = mypaper.exe
EXE __DLL = mypaper.dll
```

其中的一行即“EXE = mypaper.exe”表示用户应用程序在经过编译连接后，生成 mypaper.exe 文件，该文件供 PRO/E 启动时调用.

3) 注册用户应用程序，应用 protk.dat 文件对用户应用程序注册，protk.dat 文件的内容如下：

```
NAME mypaper
STARTUP exe
EXEC __FILE mypaper.exe
TEXT __DIR ./text
REVISION 2001
END
```

在该注册文件中，第一行为用户应用程序名，第二行指明启动方式为可执行文件方式，第三行为可执行文件的全名，第四行为本应用程序所需要的所有文本文件所在的目录，第五行为 PRO/E 的版本.

4) 加载用户应用程序后，PRO/E 的 assembly 菜单中自动添加 Assem SubCAD 菜单.

7 结论

PRO/E 作为功能强大的 CAD/CAM 系统，为用户提供了二次开发的工具 PRO/T，用户可以根据自己的需要定制自己的子系统集成到 PRO/E 中；而 PRO/T 提供了用户界面的开发接口，也可以应用 DLL 技术调用 MFC 对话框，从而为用户提供优良的交互界面.

参考文献：

[1] parametric technology corporation，Pro/Toolkit user's Guide [Z].USA:PTC 公司，2001.

[2] 黄圣杰，等.PRO/Engineer2001 高级开发实例[M].北京：电子工业出版社，2002.

[3] 北京博彦科技发展有限责任公司.Visual C++开发教程[M].北京：清华大学出版社，2000.

The Research and Application of Pro/Toolkit Development

ZHOU Hui-Lan，ZHOU Xin-jian

(School of Mechanical and Electrical Engineering，East China Jiaotong University. Nanchang 330013，China)

Abstract：In the paper，the main technology and method of Pro/Toolkit development is introduced in detail. The Pro/Toolkit in Pro/E is the application interface substantially，having great advantage in developing and customizing API for developer.

Key words：Pro/Toolkit；the development