

基于 Pro/E 二次开发的余量求解应用程序设计

李文静^{1,2} 干为民¹ 徐波¹

¹(江苏省特种加工重点实验室(常州工学院) 江苏 常州 213032)

²(常州大学机械工程学院 江苏 常州 213164)

摘要 为解决复杂零件在加工过程中余量求解难的问题,在 Visual C++6.0 环境下,选用 Pro/Toolkit 为开发工具,进行 Pro/E 二次开发。开发出一套基于 Pro/E 系统的应用程序模块,以实现设计 3D 原型与测得点云的调试和对比,及加工余量的求解。介绍 Pro/E 二次开发的方法及模式,详细阐述开发余量求解应用程序的开发过程以及程序编译。以电解加工整体叶轮叶片为实例,显示了以计算机辅助测量方法进行余量求解是提高检测精度、缩短后续电解加工周期的有效途径。

关键词 程序设计 Pro/E 二次开发 余量求解

中图分类号 TP3 文献标志码 A DOI:10.3969/j.issn.1000-386x.2020.11.004

APPLICATION PROGRAM DESIGN OF SOLVING THE ALLOWANCE BASED ON SECONDARY DEVELOPMENT OF PRO/E

Li Wenjing^{1,2} Gan Weimin¹ Xu Bo¹

¹(Jiangsu Key Laboratory of Non-Traditional Machining(Changzhou Institute of Technology), Changzhou 213032, Jiangsu, China)

²(School of Mechanical Engineering, Changzhou University, Changzhou 213164, Jiangsu, China)

Abstract In order to solve the problem of solving the allowance in the processing of complex parts, under the Visual C++6.0, Pro/Toolkit is used as the development tool to carry out the second development of Pro/E. This paper develops a set of application module based on Pro/E system, so as to realize the debugging and comparison of 3D prototype design and point cloud measurement, and the solution of machining allowance. It introduced the methods and patterns of the secondary development of Pro/E, and expounded the development process and program compilation of the application program of development allowance solving in detail. Taking the electrochemical machining of integral impeller blade as an example, it is shown that the computer-aided measurement method for solving allowance is an effective way to improve the detection accuracy and shorten the subsequent electrochemical machining cycle.

Keywords Program design Pro/E Secondary development Allowance for solving

0 引言

常用的 CAD/CAM 软件功能多样且强大,但不能满足特定用途的设计要求。若能基于此类软件进行针对产品的二次开发,将能显著提升现有的生产效率。“二次开发”就是基于类似 UG、Pro/E 的软件系统,开发出有针对性的,将软件本地化的程序的过程。目前,国内外有许多专家学者都对 Pro/E 的二次开发进行了研究。张峰^[1]以弧面状的凸轮机械手产品为研究对

象,利用 Pro/Program 模块对 Pro/E 进行二次开发,实现了产品模型的结构参数化设计,提高了设计效率和设计质量。王冰冰等^[2]研究了 Pro/E 软件二次开发的包装机械的参数化设计和实施方法,通过在系统中输入参数,可以自动生成三维实体模型和二维工程图,从而提高设计效率。刘颖等^[3]研究了 Pro/E 二次开发在圆柱齿轮减速器建模中的应用,并阐述了 Pro/Toolkit 二次开发的基本步骤,详细介绍了源文件编译,可执行文件的生成、注册和运行等关键技术。Hsu 等^[4]以 Pro/E 软件为平台,通过编程 C 语言开发了一套装配

设计分析和产品概念设计系统,系统通过五个模块实现了自动化产品。Haapala 等^[5]利用 Pro/E 建立了头骨的三维模型,并赋予其质量属性,如密度、表面积和惯性矩,然后编制了相应的 Pro/Toolkit 应用程序以标记两个肌肉集合的位置,最后创建了一个 3D 同胚扫描视眼和头部运动系统。

电解加工具有工具无损耗和加工过程无切削应力的特性,故经常采用该方式加工复杂的叶轮类零件^[6]。整体叶轮粗加工后,其叶片呈不规则曲面,与预期达到的图纸上的尺寸还有一定的差距。本文二次开发的模块程序就是为了简化测量加工后零件余量大小,从而生成误差报告,为后续的加工提供了很大的便捷。二次开发的程序不仅可以缩短测量的周期,而且大大提高测量的精度,更是后续加工过程的一个重要的环节。

1 Pro/E 二次开发方法及工作模式

Pro/E 软件是 PTC 在美国开发的 3D 设计软件,不仅具有参数化、相关化的特点,而且具有更加细致而强大的设计、分析和制造功能,应用比较广泛^[7]。其二次开发主要有基于 Pro/Toolkit、Automation GATEWAY、J-Link、Pro/Web Link、VB API 的五种二次开发方法。本文选择基于 Pro/Toolkit 的二次开发方法,该方法不仅可以自动建模,还可以集成需要访问 Pro/E 数据或操作的外部应用程序,而且可以扩展软件用户界面,让定制的流程无缝嵌入到用户界面中^[8]。

Pro/Toolkit 有异步和同步两种工作模式。异步模式的优点是 Pro/Toolkit 应用程序可以在 Pro/E 系统关闭时单独运行,但缺点是其程序代码复杂且运行速度较慢,因此通常不使用异步模式。在同步模式下,Pro/Toolkit 应用程序与 Pro/E 系统必须同步运行,并且在 Pro/E 系统关闭时无法运行。同步模式分为 DLL 和多进程模式,Pro/Toolkit 和 Pro/E 集成时采用的标准方法就是 DLL 模式^[9]。由于 Pro/Toolkit 应用程序在 Pro/E 系统上运行,所以一般采用同步模式。

2 余量求解二次开发过程

2.1 二次开发设计流程

图 1 为 Pro/E 二次开发的余量求解的设计流程图。首先,在安装好的 Pro/Toolkit 中对 DLL 文件进行创建和设置,其中包含关键的“库文件”设置,若设置不对或缺少会导致后期出错。在二次开发的整个流程中,最为关键的就是程序编译,程序编写主要分为三大

模块:导入模型、导入数据、误差分析计算。其中误差分析是本程序二次开发的核心,其目的是对导入点进行信息收集(主要是坐标属性信息),对叶片曲面进行分解(可以看成无数的点组成),比对点到叶片距离最近的点,并拾取出来,转换成点到点的距离计算,并得出测量结果。最后只要导入原设计模型和三坐标测得的云点数据,就可以直接计算出加工后零件尺寸与图纸设计尺寸的误差,同时生成误差数据表。

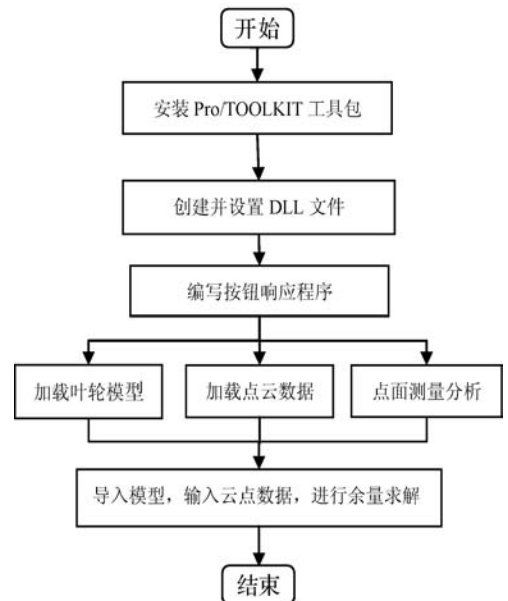


图 1 总体设计流程图

2.2 二次开发程序的准备工作

在二次开发工具包 Pro/Toolkit 安装成功的基础上,需创建和配置 DLL 文件。创建 DLL 文件需先创建一个项目 MFC DLL 文件,新建一个名称 Analytical Tools 的解决方案后完成 MFC DLL 文件的创建。然后,设置包含文件和库文件,在“配置属性”中“VC++ 目录”中添加“包含目录”,如图 2 所示。将库文件添加到“显示以下内容的目录”中,如图 3 所示,编辑“预处理器定义”,添加 PRO_USE_VAR_ARGS。最后,编辑“附加依赖库”,完成二次开发的准备工作。



图 2 添加包含目录

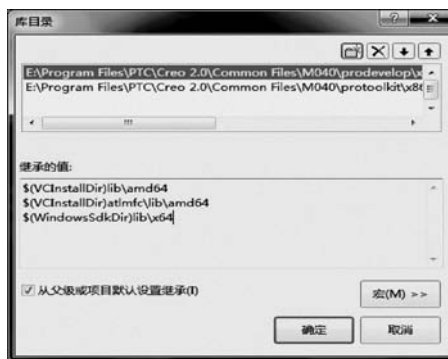


图3 添加库文件

2.3 创建菜单按钮

1) 程序头文件。在主文件中需添加 Pro/Toolkit 头文件,常用的头文件亦可添加,这样无须在添加新的 CPP 文件时设置一次头文件。如果使用了 Pro/Toolkit 对象函数,则应用程序必须包含 ProToolkit.h 头文件。

2) 菜单按钮的创建。余量求解的程序中有三个按钮,分别是“加载叶轮模型”、“添加云点数据”和“点面测量分析”。菜单按钮的创建也分为三个部分:(1) 添加菜单按钮;(2) 读取信息文本文件;(3) 添加父菜单。最后编写信息文本文件,即按钮的名称。创建完成后如图 4 所示。

```
int AnalyticalToolsmenu::menuAddexcute()
{return menuAdding();
}
//添加菜单按钮
int AnalyticalToolsmenu::menuAdding()
{
OTK::pfcUICCommand uicmd;
uicmd.SetMsgFile(_T("AnalyticalTools_msg.txt"));
//读取信息文本文件
uicmd.parent_designate("AnalyticalToolsMenubar","AnalyticalTools_Menubar","Utilities");
//添加父菜单
//信息文本文件:
AnalyticalTools_Menubar
Creo2 Wizard
#
AnalyticalToolschildmenu0
AnalyticalToolschildmenu0
加载叶轮模型
#
:
```



图4 菜单按钮的创建

2.4 导入模型模块

“加载叶轮模型”按钮的开发其实质是对原有功能“打开文件”的调用,先添加菜单按钮,再去调用已有函数按钮 ProCmdModelOpen,可以实现打开选择模型的模块。

```
uicmd.Designate("AnalyticalTools_child_action0","AnalyticalToolschildmenu0","AnalyticalToolschildmenu0-help","button.gif",AnalyticalToolschildmenu0action);
//添加菜单按钮 0
void Designate(IN CharPtr action_name,IN CharPtr Label,IN CharPtr pHelp,IN CharPtr icon,IN MyFunction func,IN MyuiCmdAccessFn acc = uiAccess)
//对 Designate 的定义
//菜单按钮 0 的主函数:
```

```
extern ProError AnalyticalToolsmenu::AnalyticalToolschildmenu0action()
{
return ProMacroLoad(L"~Command 'ProCmdModelOpen'");
//调用函数按钮 ProCmdModelOpen 使用“摘要”样式
}
```

2.5 导入点云文件模块

导入点云文件模块其实质和打开模型模块的实质相似,是对按钮“导入”的调用,故与打开模型模块程序编写类似,只是改变要调用的函数按钮。

```
extern ProError AnalyticalToolsmenu::AnalyticalToolschildmenu1action()
{return ProMacroLoad(L"~Command 'ProCmdModelImpAppendNew'");
}
//调用已有函数按钮
ProCmdModelImpAppendNew
```

2.6 余量求解模块

余量求解模块是本次二次开发程序的核心,依旧是先创建菜单按钮,这与先前模块一致,再对菜单按钮的主要函数进行编写这是关键一步。主要分为以下几个部分:

(1) 对输入的 pts 云点数据文件进行添加:

```
ProError getfeats(ProFeature p_feat,ProError status,ProAppData app_data)
{ProArrayObjectAdd((ProArray*)app_data,PRO_VALUE_UNUSED,1,&p_feat);
//将指定位置的对象添加到数组中,或者将一个对象追加到数组的末尾
```

```
return(status);}
```

(2) 对输入点进行特征收集:

```
ProError getfeats(ProFeature p_feat,ProError status,ProAppData app_data)
{ProArrayObjectAdd((ProArray*)app_data,PRO_VALUE_UNUSED,1,&p_feat);return(status);}
extern ProError ProArrayObjectAdd(ProArray* p_array,int index,int n_
```

objects,void * p_object); For example, if you want to insert one integer in an integer array, call the function as follows: ProArrayObjectAdd (&int_array, 0, 1, (void *)&int_val);

(3) 收集模型上云点投影的基准点:

```
ProError getGeoms(ProGeomitem * geom,ProError err,ProAppData data)
{ err = ProArrayObjectAdd((ProArray *)data, PRO_VALUE_UNUSED, 1, (void *)geom);
return err;
}
ProError featGeomsCollect (ProFeature * feat, vector <ProGeomitem > &p_POINT_data)
{ProError status;
ProFeature * p_data;
Int nsize = 0;
status = ProArrayAlloc(0, sizeof(ProGeomitem), 1, (ProArray *)&p_data);
status = ProFeatureGeomitemVisit (feat, PRO_POINT, (ProGeomitemAction)getGeoms, NULL, (ProAppData)&p_data);
//对项目进行编译
```

```
ProArraySizeGet((ProArray)p_data, &nsize);
for(int i=0; i < nsize; i++)
{p_POINT_data.push_back(p_data[i]);}
return ProArrayFree((ProArray *)&p_data);
}
```

(4) 对话框显示数据:

```
class SolidOutlineUI
public: static void UI_Create()
{ char * dialog_name = "SolidOutline";
int ActiveDialog_status;
ProError err;
err = ProUIDialogCreate(dialog_name, dialog_name);
```

(5) 对固定面 ID 的拾取:

```
ProError status;
ProSelection surface_sel, cancel;
ProModelitem surface_item; //信息过滤函数
surface_item.id = 11626; //面 ID
surface_item.type = PRO_SURFACE; //模型曲面
surface_item.owner = Curmdl;
```

(6) 如果不成功手动选取面:

```
status = ProSelectionAlloc(NULL, &surface_item, &surface_sel); if (status != PRO_TK_NO_ERROR) {ProError err;ProSelection * p_Selection; int n_sels; err = ProSelect("surface", 1, NULL, NULL, NULL, NULL, &p_Selection, &n_sels); if (err != PRO_TK_NO_ERROR || n_sels == 0)
{return;} ProSelectionCopy(p_Selection[0], &surface_sel); } vector <ProGeomitem > p_POINT_data; for (int i = 1; i < p_Feature_data.size(); i++) P{featGeomsCollect(&p_Feature_data[i], p_POINT_data);} double distance;
```

```
Pro2dPnt uv1, uv2;
Pro3dPnt xyz1, xyz2;
```

(7) 对应面的最短距离坐标显示:

```
CString strHead[] = {L"面坐标", L"点坐标", "测量值"};
int Widths[] = { 11, 11, 7 };
TableHeadSet(dialog, "Table1", p_POINT_data.size(), 3, strHead, Widths); for(int i = 0; i < p_POINT_data.size(); i++) {status = ProSelectionAlloc(NULL, &p_POINT_data[i], &cancel); if (status == PRO_TK_NO_ERROR)
{status = ProSelectionDistanceEval(surface_sel, cancel, uv1, uv2, xyz1, xyz2, &distance);
//计算两个点之间的距离
```

(8) 对话框输出表格建立:

```
private:static ProError TableItemTextSet(IN char * dialog_name,
IN char * TitleName,
IN int row,
IN int column,
IN CString ItemText)
{
ProCharLine strtmp, strtmp1;
sprintf_s(strtmp, "col% d", column);
sprintf_s(strtmp1, "row% d", row);
return ProUITableCellLabelSet(dialog_name, TitleName, strtmp1, strtmp, (wchar_t*)(LPCTSTR)ItemText);}
//设置表单元格中包含的文本
```

(9) 对话框表格的划分:

```
for(int i = 0; i < column; i++)
{names[i] = (char *) calloc(PRO_NAME_SIZE, sizeof(char));
labels[i] = (wchar_t *) calloc(PRO_PATH_SIZE, sizeof(wchar_t));
sprintf_s(strtmp, "col% d", i + 1); strcpy(names[i], strtmp);
wcsncpy(labels[i], (wchar_t*)(LPCTSTR)pHead[i]);
}
status = ProUITableColumnnamesSet(dialog_name, TitleName, column, names);
//表格分列
status = ProUITableColumnlabelsSet(dialog_name, TitleName, column, labels);
//设置表格中每列标签的名称
ProUITableColumnwidthsSet(dialog_name, TitleName, column, pHeadWidths);
//设置表格中每列的宽度
:
ProUITableShowgridSet(dialog_name, TitleName, pGrid);
return status; } }
```

//表格显示

2.7 其他辅助程序模块

最后,还需编写一些辅助程序如注册文件程序和菜单图标程序。菜单图标如图 5 所示。

```

NAME AnalyticalTools
startup dll
EXEC_FILE AnalyticalTools_dir\PRO_MACHINE_TYPE\AnalyticalTools.dll
TEXT_DIR AnalyticalTools_dir\text
allow_stop true
DELAY_START false
revision Wildfire
END //注册文件
Button.gif //菜单图标文件

```



图 5 菜单图标

3 程序的配置与运行

在以上设置完成基础上,再在“资源管理器”项目中生成程序,生成程序如图 6 所示,然后对程序进行调试,调试结果如图 7 所示。

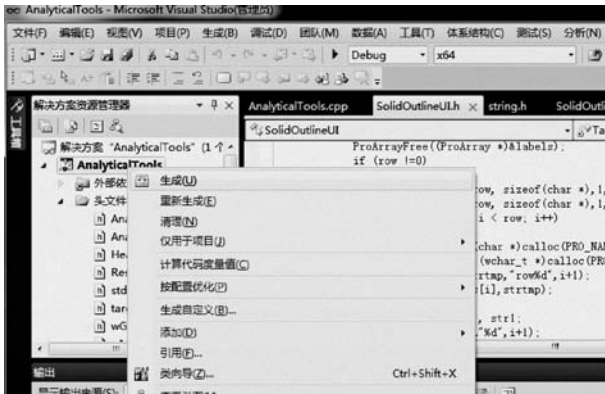


图 6 生成程序

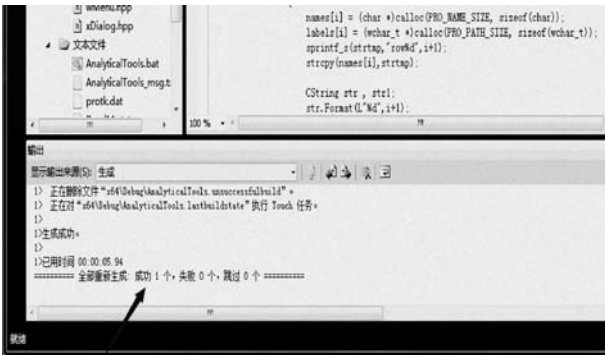


图 7 程序调试成功

调试成功后需进行创建快捷启动方式,创建程序如下:

```
@ echo off
```

```

set AnalyticalTools_dir = . . .
set lang = chs
start "" "E:\Program Files\PTC\Creo 2.0\Parametric\bin\parametric.exe"
Exit //创建快捷方式

```

4 电解加工整体叶轮叶片余量求解实例

4.1 三坐标机采集点云坐标

运用三坐标测量机进行数据采集,得到点云数据,对粗加工后叶轮叶片用三坐标机进行划线测量取点,如图 8 所示,得到点云坐标。由于人为或随机因素在测量过程中的影响,测量中存在不稳定噪声点,文件需要点转换,并且去除不满足的点。对云点坐标进行处理并转换成可以导入 Pro/E 点文件. pts 格式文件,转换后的. pts 格式点文件如图 9 所示。

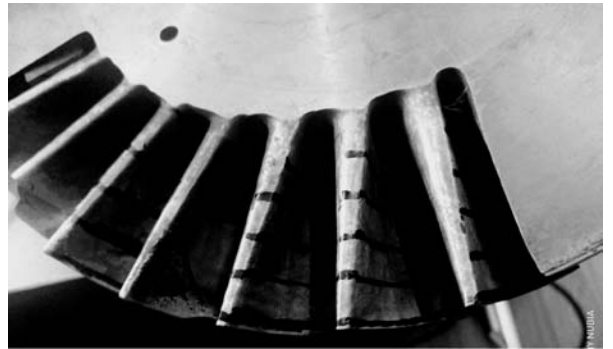


图 8 划线测量

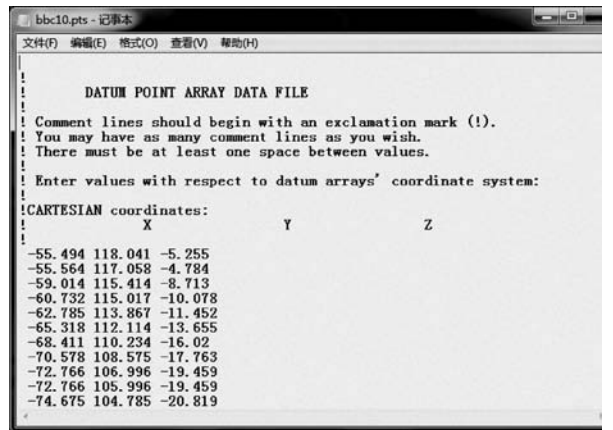


图 9 处理后点云数据

4.2 余量求解实际应用

针对电解加工后的整体叶轮叶片进行余量求解,打开“工具”菜单栏,子菜单为二次开发程序菜单,包含“加载叶轮模型”“加载点云数据”“点面测量分析”三个按钮。第一步,点击按钮“加载叶轮模型”,加载完成后如图 10 所示。第二步,点击“加载点云数据”即导入之前的. pts 格式的点云文件,加载完成后如图

11 所示。第三步,点击“点面测量分析”,由于一个叶片是由至少两个曲面组成,所以叶轮叶片存在正面与反面的误差求解。选择需要测量的曲面,叶片正反面余量求解的结果如图 12 所示。



图 10 加载叶轮模型

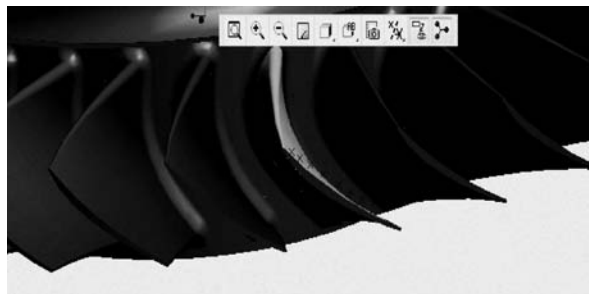
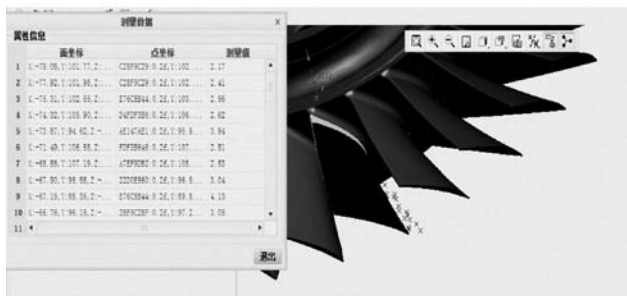


图 11 加载点云数据



(a) 正面



(b) 反面

图 12 整体叶轮叶片余量求解结果

4.3 余量求解结果分析

一般误差分析方法,针对简单的有规则形状零件可以直接用现有的测量工具测量,但数据不精确,而复杂的零件如整体叶轮叶片之类曲面结构较多的零件会

利用算法来进行测量,工作量较大且易出错。相比之下,二次开发余量求解程序响应快,只需很短的时间就可以得到结果并且不易出错。

5 结 语

本文以 Pro/E2.0 为平台,基于开发语言 VC ++ 6.0,以 Pro/Toolkit 为二次开发工具,使用编译平台 VS2010,开发出一套能在 Pro/E 中实现粗加工模型到设计模型的余量计算的应用程序,并成功地应用在电解加工整体叶轮中,代替传统的误差分析方法,大大缩短测量周期,提高加工效率,降低生产成本,成为后续加工设计的重要组成部分。

参 考 文 献

[1] 张锋. 基于 Pro/E 二次开发的机械产品的三维参数化设计[J]. 机械设计与制造, 2007(5) :38 - 39.

[2] 王冰冰,林木,赵书强. 基于 Pro/E 二次开发的包装机械参数化设计与实现[J]. 机械制造与自动化, 2017, 46(2) : 101 - 103.

[3] 刘颖,彭雪鹏. Pro/E 二次开发技术在圆柱齿轮减速器建模中的应用[J]. 机械设计与制造, 2006(3) :86 - 88.

[4] Hsu W, Fuh J Y H, Zhang Y. Synthesis of design concepts from a design for assembly perspective [J]. Computer Integrated Manufacturing Systems, 1998, 11(1 - 2) : 1 - 13.

[5] Haapala S A, Enderle J D. Dynamic modeling of the neck muscles during horizontal head movement. Part II: Model construction in Pro/Engineer [J]. Biomedical Sciences Instrumentation, 2003, 39:71 - 76.

[6] 朱永伟,徐家文,胡平旺. 数控展成电解加工整体叶轮的研究与应用[J]. 航空学报, 2001, 22(4) :376 - 378.

[7] 王思远,贺成柱. 基于 Pro/Toolkit 的 Pro/E 工程图标注系统的二次开发研究[J]. 甘肃科技, 2016, 32(12) :18 - 22.

[8] 刘玉秦,宋彦. 基于 Pro/TOOLKIT 的 Pro/E 二次开发技术研究[J]. 制造业自动化, 2011, 33(10) :6 - 9.

[9] 孙锐,钟佩思,刘梅,等. 基于 Pro/E 二次开发的特征提取技术研究[J]. 机械设计与制造, 2012(4) :86 - 88.

(上接第 10 页)

[13] Liu C, Chen C, Han J W, et al. GPLAG: Detection of software plagiarism by program dependence graph analysis [C] // Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006 :872 - 881.

[14] Nguyen H A, Nguyen T T, Wilson G, et al. A graph-based approach to API usage adaptation [J]. ACM Sigplan Notices, 2010, 45(10) :302 - 321.