

## GRIP、API 与 KF 建立零件模型方法之比较

金丽华

(盐城技师学院 机械系,江苏 盐城 224002)

**摘要:**UGNX 提供了 GRIP、NX/OpenAPI、KF 等开发工具和接口让客户捕捉和保留领域知识,包括:捕捉流程知识、自动化设计和定制用户使用环境等。从人机界面、源程序代码及其执行结果等方面比较了这3种方法在建立零件模型时的优缺点,为二次开发人员选择开发工具提供帮助。

**关键词:**GRIP;API;KF;建模

**中图分类号:**TP391.73 **文献标识码:**A **文章编号:**1671-5322(2011)04-0030-04

UGNX 软件强大的功能,特别是虚拟装配的直观性,使得设计人员在产品制造前即可对其性能进行评价,做到及时发现问题并修改,由此极大地提高了设计效率和水平。然而在产品的设计过程中,零件建模工作量太大,严重制约着设计工作的效率,例如工装设计中的标准件-螺钉等,由于规格的不同需要多次重复建模,这样经常性的重复劳动让许多设计员感到厌烦。在这种情况下,迫切需要通过 UG 的二次开发,建立产品的通用件、标准件、常用组装的零件库以供调用。

本文主要介绍采用 GRIP 语言、UF 以及 KF3 种不同的方式建立标准件-垫片的方法,比较这3个方式开发零件库的优缺点。

### 1 GRIP 方法

#### 1.1 人机界面

在 GRIP 语言中,用户可利用 GRIP 程序中的人-机交互语句与 UG 系统对话,完成人机操作。常用的 GRIP 交互语句有:pos、gros、choose、mchoose、param、ident、text、messg 等语句。本例中的人机界面采用 param 语句实现。

#### 1.2 程序代码

```
entity/cyl(3)
DATA/D0,20,D1,15,H,3
lab1:
param/please enter', $
```

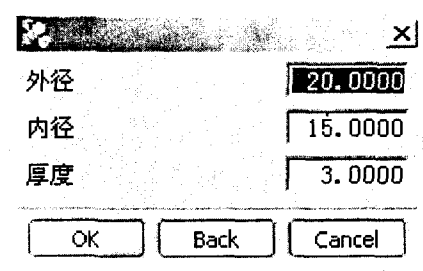


图1 垫片对话框

Fig. 1 Gasket dialog

外径,D0,\$

内径,D1,\$

厚度,H,resp

jump/lab1:,stop:,,,resp

cyl(1) = solecyl/origin,0,0,0,height,H,dia-  
mtr,D0

cyl(2) = solecyl/origin,0,0,0,height,H,dia-  
mtr,D1

cyl(3) = subtra/cyl(1),with,cyl(2)

stop:

halt

GRIP 程序一般在 GRADE (GRIP 高级开发环境) 中进行编译、链接、生成可执行的 GRIP 程序。进入 GRADE 的方法是在 Windows 环境下。开始→程序→Unigraphics NX2.0→Unigraphics Tools→UG Open GRIP。在 UG 环境下,执行 GRIP 程序有多种方法,主要有:(1)选择 File→Execute

收稿日期:2011-11-14

作者简介:金丽华(1973-),女,江苏阜宁人,讲师,主要研究方向为公差配合与模具设计。

UG/Open→ Grip 在“执行 GRIP 程序”的对话框中选择编译链接后的 GRIP 程序;(2)菜单调用;(3)用户工具调用。本例通过方法一调用。

### 1.3 执行结果

GRIP 程序的执行结果如图 2 部件导航器所示,垫片通过两个圆柱体的布尔操作生成,故该模型不具有相关性,即模型在 UG 环境下不便于编辑修改。如需修改垫片参数则需重新运行 GRIP 程序。

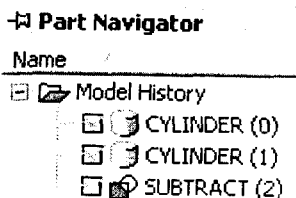


图 2 部件导航器

Fig.2 Component navigator

## 2 UF (user function) 方法

### 2.1 人机界面

采用 UF 方法开发时,人机界面主要通过 UG/OPEN UI Styler 模块来定制。该模块不光提供强大的制作 UG 风格窗口的功能,同时还生成 C 程序模板框架,使程序开发工作量有所减少。

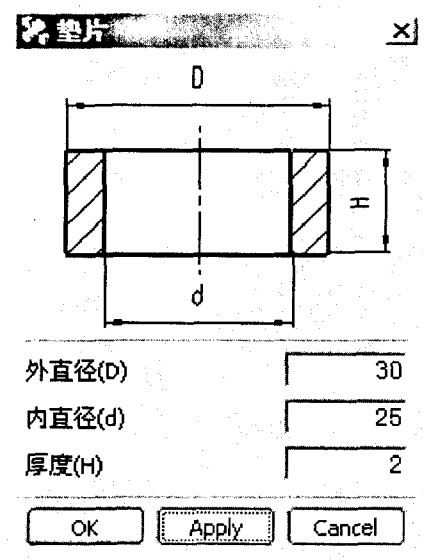


图 3 垫片对话框

Fig.3 Gasket dialog

### 2.2 部分程序代码

```
void UG_OPEN_create_pad(double d1, double
```

```
d2, double h)
```

```
{
    UF_FEATURE_SIGN sign = UF_NULLSIGN;
    double origin[3] = {0, 0, 0};
    char height[50], diam[50];
    double direction[3] = {0, 0, 1};
    tag_t cyl_obj_id;
    uf_list_t * edge_list, * face_list;
    tag_t * edge_array, * face_array;
    int edge_count, face_count;
    double h_location[3] = {0, 0, 0}, h_direc-
tion[3] = {0, 0, -1};
    char diame[50], depth[50], angle[50];
    tag_t hole_tag;
    int rc;
    sprintf(height, "h = %f", h);
    sprintf(diam, "D = %f", d1);
    h_location[2] = h;
    sprintf(diame, "D1 = %f", d2);
    sprintf(depth, "%f", 0.0);
    sprintf(angle, "%f", 0.0);
    rc = UF_MODL_create_cyl1(sign, origin,
height,
diam, direction, &cyl_obj_id);
    if(! rc)
    {
        UF_MODL_ask_feat_edges(cyl_obj_id, &edge
_list);
        UF_MODL_ask_feat_faces(cyl_obj_id, &face_
list);
        UF_MODL_ask_list_count(edge_list, &edge_
count);
        edge_array = (tag_t *) malloc(sizeof(tag_
t) * edge_count);
        UG_OPEN_list_tras_array(edge_list, edge_ar-
ray);
        UF_MODL_ask_list_count(face_list, &face_
count);
        face_array = (tag_t *) malloc(sizeof(tag_t)
* face_count);
        UG_OPEN_list_tras_array(face_list, face_ar-
ray);
        rc = UF_MODL_create_simple_hole(h_loca-
tion, h_direction, diame, depth, angle,
```

```

face_array[0], face_array[1], &hole_tag);
UF_free(edge_list);
UF_free(face_list);
free(edge_array);
free(face_array);
}
}

void UG_OPEN_list_tras_array(uf_list_t *obj
_list, tag_t *obj_array)
{
while(obj_list != NULL)
{
*obj_array++ = obj_list->eid;
obj_list = obj_list->next;
}
}

```

UF 程序一般在 VC++ 集成开发环境中进行编译、链接、生成 DLL(内部模式)。在 UG 环境下,执行 UF 程序有多种方法,主要有:(1)选择 File→Execute UG/Open→User Function,在“执行 User Function 程序”的对话框中选择 DLL,通过这种方式调用的 UF 程序的入口函数为 ufusr;(2)菜单调用,通过这种方式调用的 API 程序的入口函数为 ufsta。本例通过菜单调用,故需在相应菜单项的 actions 语句后加对话框文件名实现菜单对 API 程序的调用。

### 2.3 执行结果

UF 程序的执行结果如图 4 部件导航器所示,生成的垫片模型是参数化的。在 UG 环境下,可以通过编辑表达式或特征的方式对其编辑修改,即可通过更改零件的几何参数编辑零件模型。

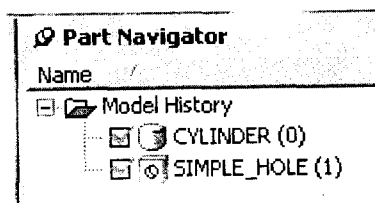


图 4 部件导航器

Fig.4 Component navigator

## 3 KF(Knowledge Fusion)方法

采用 KF 方法时,人机界面也可通过 UG/OPEN UI Styler 模块来定制。故本方法的人机界

面创建方式与 UF 方法相同。需要注意的是对话框文件名、KF 文件名与类名必须相同,对话框中控件的 ID 号与 KF 程序中的属性名必须相同。

### 3.1 程序代码

```

#! UG/KBE 17.0

Defclass: pad(ug_base_part);
(integer Parameter) d_max: 30;
(integer parameter) d_min: 25;
(integer parameter) h: 2;
(Child) out_cylinder: {Class, ug_cylinder, diameter, d_max:, height, h:};
(list) c_faces: ug_feature_askFaces( out_cylinder: );
(list) c_edges: ug_feature_askEdges( out_cylinder: );
(child) in_c: { class, ug_simple_hole, hole_diameter, d_min:, placement_face, { nth (2, out_cylinder: c_faces: ) }, thru_face, { nth (1, out_cylinder: c_faces: ) }, rpo, { nth (1, out_cylinder: c_edges: ), RPO_ARC_CENTER, 0, RPO_ARC_CENTER, 0, RPO_PARA_DIST } }, };

```

KF 程序是面向对象的编程语言,支持对象的分类、继承;同时它也是解释性语言,不需要编译。在菜单文件中,只需在 actions 后直接加 KF 文件名即可实现菜单对 KF 程序的调用。

### 3.2 执行结果

由 KF 程序生成的垫片,在图 5 的知识融接导航器中可知,它以子规则形式存在。同时在图 6 的部件导航器中,其快捷菜单中有 EDIT KF Application Object 菜单项,选择该菜单项后出现的编辑窗口与创建垫片的对话框相同。这样的模型不但具有相关性,而且在编辑修改时还屏蔽了垫片中的几何要素,使垫片的编辑只与其工程技术参

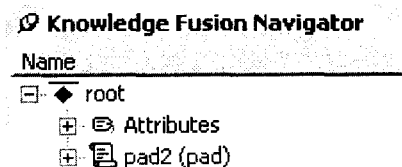


图 5 知识融接导航器

Fig.5 Knowledge fusion navigation device

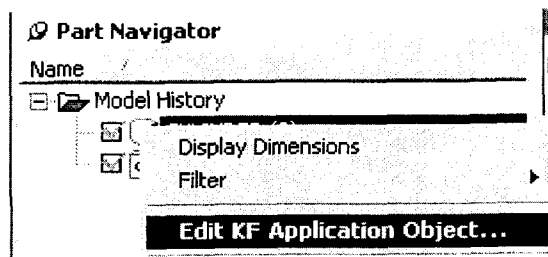


图6 部件导航器

Fig.6 Component navigator

数相关。

#### 4 总结

GRIP具有简单、方便的特点,工程师只要具

有初步的编程知识,即可上手,但由GRIP程序生成的模型无相关性,不便编辑修改;UF包含了约2 000个C函数,可实现绝大部分的UG操作,通过调用这些C函数,用户可以建立、编辑、查询UG的各种实体对象,由UF生成的模型具有相关性,可以在UG环境下自由地编辑修改。在这3种方法中,UF实现的功能最全,但其较为复杂,不易上手;KF语言主要来源于Heide公司开发的Intent语言,并经过长期的改进,现已成为功能强大的基于知识的工程设计语言,作为一种解释性自然语言,其应用程序无须编译即可执行。由KF生成的模型,无论是创建还是编辑都只需设计人员输入工程参数,即可由知识驱动几何模型的变化,是CAX技术发展的新阶段。

#### 参考文献:

- [1] 李立英. 基于反求工程的吹塑模具CAD系统开发[D]. 南昌:南昌大学,2006.
- [2] 唐文献,李莉敏,谢心意. 产品自组织进化创新中的本体知识体系[J]. 机械设计与制造,2005(11):77-81.
- [3] 黄勇. 基于UG的齿轮刀具设计及切齿仿真加工研究[J]. 食品与机械,2006(11):91-94.

## A Comparison of GRIP, API and KF in Creating Modeling

JIN LI - hua

(Department of Mechanical Engineering of Yancheng Technicians College, Yancheng Jiangsu 224002, China)

**Abstract:** UCII provides GRIP, NX / Open API, KF and other development tools and interfaces that allow customers to capture and retain domain knowledge, including: capture process knowledge, automation design and customize the user environment. This paper compares these three methods in part modeling from the perspective of man - machine interface, source code and its implementation result, which provides a reference for the selection of the secondary development tools.

**Keywords:** GRIP; API; KF; Modeling

(责任编辑:沈建新)