SolidWorks 二次开发在工程图属性中的应用

河南科技大学材料科学与工程学院 （洛阳 471003） 刘 毅 辛选荣 刘 汀

一、问题的提出

SolidWorks 是一个全相关的三维机械设计软件，它将三维造型技术与二维绘图融为一体，模型文件中的设计数据如尺寸、注释、文件属性等内容与工程图相关，这样的处理方法不仅保证了设计信息的一致性，而且大大提高了设计效率。但 SolidWorks 软件从模型直接生成的工程图所标注的尺寸、字体及注释都不太尽如人意，用软件本身的注解工具一个一个添加或修改显得有些烦琐，能否通过对 SolidWorks 二次开发建立一个单独的工作界面，可以很方便地添加和修改这些信息就成了问题的关键。

二、工程图属性

为了更大程度地提高工作效率，我们利用 VB6.0 通过对 SolidWorks 二次开发建立了工程图属性程序，这个程序通过一个对话界面直接对当前的工程图文件尺寸、字体、视图、注解等属性进行添加或修改。本程序利用 VB6.0 进行编译生成可执行文件，其界面如图 1 所示。



图 1 工程图属性程序

这个程序的基本功能如下。

1. 尺寸方面

可以自动插入模型注解尺寸，如果自动插入的尺寸不太合适，也可以强制修改自动插入的尺寸，同时还可以修改尺寸公差。

2. 比例及高度

选中工程图上的一个视区后，可以改变所选视区的比例大小，也可以改变工程图注解尺寸、字体的高度及高宽比例。

3. 注释

在指定的位置可以自动添加常用的粗糙度符号、形位公差符号、基准代号、注释等内容。

该程序是利用 SolidWorks 提供的 API 函数对当前的工程图文件进行修改的操作，下面介绍一下程序中用到的 SolidWorks API 对象和方法。

（1）获得 Solidworks 工程图对象 在 VB 中利用 CreateObject 方法来获得 SolidWorks 对象，Activedoc 方法可以获得文件对象，利用 GetType 方法获得当前文件的类型，如果文件对象为空或者是模型文件与装配体文件时就会弹出提示对话框，提示用户打开一个工程图文件。

```
Set sw = CreateObject ( "SldWorks.Application")
Set activeDocument = sw.Activedoc
docType = activeDocument.GetType
Dim Msg, Style, title
Style = vbExclamation
title = "工程图属性"
Msg = "请打开一个工程图文件"
If (activeDocument Is Nothing) Then
Call MsgBox (Msg, Style, title)
End
ElseIf (docType = swDocPART) Then
Call MsgBox (Msg, Style, title)
End
ElseIf (docType = swDocASSEMBLY) Then
Call MsgBox (Msg, Style, title)
```

```
End
End If
```

（2）自动插入尺寸　通过点击自动插入尺寸按钮，可以将模型中的注解尺寸插入到工程图中。

```
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
boolstatus = Part. InsertModelAnnotations2 (0, _
True, 0, True, True, False)
```

（3）强制修改尺寸　在对话框中输入数值，可以将选中的自动插入的尺寸修改，但不能改变模型的真实尺寸。

```
Dim EditDimension As Double
EditDimension = Text1. Text
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
boolstatus = Part. Extension. SelectByID ( "", _
"DIMENSION", 0, 0, 0, False, 0, Nothing)
Part. EditDimensionProperties2 0, 0, 0, "", "", 1, _
9, 2, 1, 11, 11, EditDimension, "", 0, "", "", 0
```

（4）修改尺寸公差　选中一个带有公差的尺寸，在对话框中输入上下公差值，点击修改尺寸公差按钮，所选尺寸的公差随之修改。

```
Dim tolMin, tolMax As Double
tolMin = Text2. Text/1000
tolMax = Text3. Text/1000
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
boolstatus = Part. Extension. SelectByID ( "", _
"DIMENSION", 0, 0, 0, False, 0, Nothing)
Part. EditDimensionProperties2 2, tolMin, _
tolMax, "", "", 1, 9, 2, 1, 11, 11, _
"< MOD - DIAM >", "", 1, "", "", 0
Part. ClearSelection2 True
```

4. 比例及视图

（1）改变所选视图比例　选中一个视区，在对话框中输入合适的比例大小，代码中的 n 代表比例系数。

```
Set sw = CreateObject ( "SldWorks. Application") Set Part =
sw. ActiveDoc
Set SelMgr = Part. SelectionManager ()
Set View = SelMgr. GetSelectedObject2 (1)
currentscale = View. ScaleDecimal
n = Text4. Text
newscale = n * currentscale
View. UseSheetScale = False
View. ScaleDecimal = newscale
Part. EditRebuild
```

（2）改变字体大小　在对话框中输入合适数值。

```
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
Dim i As Integer
For i = 0 To 5
Set TextFormat = _
Part. GetUserPreferenceTextFormat (i)
TxtHeightfactor = Text5. Text/ (1000/3.5)
TextFormat. CharHeight = TxtHeightfactor
reval = Part. SetUserPreferenceTextFormat (_
i, TextFormat)
Next
Part. EditRebuild
```

（3）改变字体高宽比例　在对话框中输入合适数值。

```
Set sw = CreateObject ( "SldWorks. Application")
Set Part = swApp. ActiveDoc
Dim i As Integer
a = Text6. Text
For i = 0 To 4
Set TextFormat = _
Part. GetUserPreferenceTextFormat (i)
TxtWidthfactor = a
TextFormat. Widthfactor = TxtWidthfactor
reval = Part. SetUserPreferenceTextFormat (_
i, TextFormat)
Next
Part. EditRebuild
```

5. 注解

（1）粗糙度　在指定位置插入常用的粗糙度数值。这段代码演示的是粗糙度为 0.1 时的情况。

```
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
boolstatus = Part. Extension. SelectByID ( "", _
"EDGE", 0, 0, 0, False, 0, Nothing)
Part. InsertSurfaceFinishSymbol2 1, 0, 0, 0, 0, 0, _
1, "", "", "", "", "0.1", "", ""
```

（2）形位公差　在指定位置插入常用的形位公差，这段代码演示的是插入平行度形位工差的情况，其中 a 输入数值，b 输入基准。

```
Set Gtol = Part. InsertGtol ()
If Not Gtol Is Nothing Then
Gtol. SetFrameSymbols2 1, "< IGTOL - PARA >", _
0, "", 0, "", "", "", ""
Gtol. SetFrameValues 1, a, "", b, "", ""
Gtol. SetFrameSymbols2 2, "", 0, "", 0, "", "", _
"", ""
Gtol. SetFrameValues 2, "", "", "", "", "", ""
Gtol. SetPTZHeight "", 0
Gtol. SetCompositeFrame 0
Gtol. SetBetweenTwoPoints 0, "", ""
Set Annotation = Gtol. GetAnnotation ()
```

（3）基准代号　在指定位置产生基准代号。

```
Set sw = CreateObject ( "SldWorks. Application")
Set Part = sw. ActiveDoc
boolstatus = Part. Extension. SelectByID ( "", _
"EDGE", 0, 0, 0, False, 0, Nothing)
Dim DatumTag As Object
Dim Annotation As Object
Set DatumTag = Part. InsertDatumTag2 ()
```

```
If Not DatumTag Is Nothing Then
DatumTag.FilledTriangle = 1
DatumTag.Shoulder = 0
Set Annotation = DatumTag.GetAnnotation ()
End If
Part.WindowRedraw
Part.EditSketch
```

（4）注释　在指定的位置生成文字注释，在文本框中输入想要插入的注释内容。

```
Dim a As String
Set sw = CreateObject ( "SldWorks.Application")
Set Part = sw.ActiveDoc
a = Text9.text
Set Note = Part.InsertNote (a)
If Not Note Is Nothing Then
Note.Angle = 0
boolstatus = Note.SetBalloon (0, 0)
Set Annotation = Note.GetAnnotation ()
If Not Annotation Is Nothing Then
longstatus = Annotation. SetLeader2 (False, 0, _
True, False, False, False)
boolstatus = Annotation.SetTextFormat (0, _
True, TextFormat)
End If
End If
Part.ClearSelection2 True
Part.WindowRedraw
```

三、运行程序

SolidWorks 不能直接将 VB 编译的 exe、dll 文件作为插件使用，如果要在 SolidWorks 中直接调用通过 VB 编译的可执行程序，可以使用 SolidWorks "宏" 操作来进行。利用 "宏" 在 SolidWorks 界面中添加指定的应用程序后，就好像 SolidWorks 调用了自身的功能一样，利用类似的方法，可以在 SolidWorks 添加任何可执行的文件。下面说明一下步骤。

1. 建立 "宏" 文件

（1）选择菜单 "工具→宏操作→录制"

（2）停止录制，并保存宏文件。给定文件名称，如：工程图属性 .swp

（3）选择菜单 "工具→宏操作→编辑"，输入下面的内容：

```
Sub main ()
My AppID = Shell ( "D: \ Program Files \ SolidWorks \ 宏 \ 工程图属性 .exe", 1)
AppActivate MyAppID
End Sub
```

（4）保存并退出宏编辑，返回到 SolidWorks 环境。

2. 指定 "宏" 操作

在 SolidWorks 中可以指定宏命令的文件位置，并定制工具栏，将所执行的宏操作按钮添加到指定的位置上。选择菜单 "工具→自定义→宏"，在 "自定义" 对话框中单击 "命令" 标签，在类别中选择宏，拖动此按钮到任意标题栏，出现自定义宏按钮对话框，如图 2 所示。



图 2　自定义宏按钮对话框

单击确定，新的宏按钮就出现在工具栏中。

3. 执行应用程序

定义 "宏" 以后，单击相应的图标就可以在 SolidWorks 中直接执行应用程序，如图 3 所示。

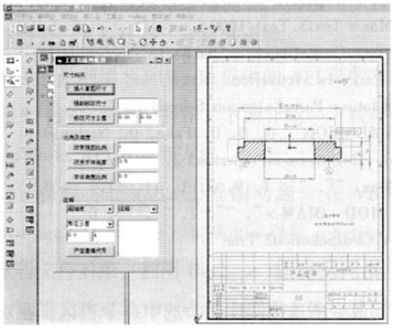


图 3　执行程序示意图

四、结论

利用 "工程图属性" 程序，可以对尺寸、字体、注释等进行快速修改，明显提高了工作效率，方便了对工程图文件的后处理工作。需要说明的是，SolidWorks API 的对象和方法十分丰富，SolidWorks 提供了几百个 API 应用程序接口函数，这些 API 函数是 OLE（对象的嵌入与链接）或 COM（组件对象模型）接口，用户可以使用 VB \ VBA \ VC \ Delphi 等多种高级语言对 SolidWorks 进行二次开发，从而建立适合用户需要的、专用的 SolidWorks 功能模块。

（收稿日期：20051230）