SOLIDWORKS 2005

Réalisation d'une macro feature

Problématique

Comment exécuter une fonction automatisée (macro) à chaque modification ou reconstruction d'un document pièce ou assemblage.

Solutions

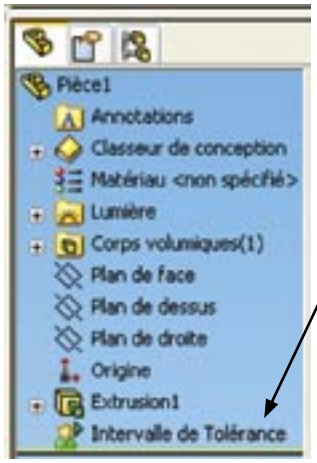
Les macros feature s'exécutent à chaque modification ou reconstruction d'un document SolidWorks

1) Principe général :

Le principe des macros feature est de s'insérer dans l'arbre de création lors de leur première exécution.

Elles deviennent ainsi totalement synchronisées avec le document, elles pourront ainsi se mettre à jour en suivant les évolutions de conception de la pièce ou de l'assemblage (mise à jour de masse, calcul de centre de gravité, etc.…).

Une macro feature apparaît dans l'arbre de construction comme une fonction de SolidWorks, elle peut être repositionnée, éditée, supprimée …



Macro feature insérée dans l'arbre de création.

Attention :

La macro feature se place dans l'arbre de création, mais le fichier source de la macro (.sldswp) n'est pas encapsulé dans le document, seul l'emplacement de la macro est mémorisé. Par conséquent, si on distribue le fichier SolidWorks sans la macro, celle-ci se met en erreur à l'ouverture du document. La solution est de placer la macro dans le même répertoire que le fichier SolidWorks et les déplacer ensemble, ou positionner les macros dans un répertoire fixe d'un serveur.

2) Réalisation d'une macro feature (voir exemple de macro «intervalle de tolérance») :

Principe de réalisation d'une macro feature :

Pour une macro standard, à chaque exécution le code démarre du «point d'entrée» de la macro, ce point d'entrée est composé du nom du module et du nom de la procédure (ce qui vous est demandé par SolidWorks si vous définissez un icône de raccourci dans une barre d'outils) Ex : «Macro1 : Main»

Dans le cas d'une macro feature, il faudra définir quatre points d'entrées :

– pour la première exécution de la macro qui n'est pas encore encapsulée dans l'arbre,
– pour la partie du code à exécuter à chaque Reconstruction,
– pour la partie du code à exécuter à chaque Edition,
– pour la partie du code à exécuter en cas de suppression.

Dans l'exemple joint à ce document, c'est la procédure «Main» du module «mldProgramme» qui s'exécute la première fois pour insérer la macro feature dans l'arbre de création.

Dans cette procédure on spécifie les autres points d'entrées qui contiendront le code à exécuter en cas de reconstruction, édition ou suppression.

L'exemple de macro feature traité dans ce document est téléchargeable sur le site www.MyCADservices.fr rubrique macro, «Intervalle de tolérance.swb»

Création de la Macro Feature :

```
Sub main()

'Déclaration des objets SolidWorks et document
'Déclaration des objets indispensables à la définition des MacroFeatures
Dim swFeatureMgr        As SldWorks.FeatureManager
Dim stFctName           As Variant
Dim vrFctType           As Variant
Dim vrFctValue          As Variant
Dim stMacroMethods(8)   As String
Dim vrMacroMethods      As Variant
Dim stFileName          As String
Dim lgOption            As Long

Set swapp = Application.SldWorks
Set swdoc = swapp.ActiveDoc

' On teste les conditions de démarrage
If swdoc Is Nothing Then
   swapp.SendMsgToUser2 "Il n'y a pas de document actif.",
   swMbWarning, swMbOk
   Exit Sub

End If
```

```
' DEFINITION DES PROCEDURES A EXECUTER EN FONCTION DES EVENEMENTS
stFileName = swapp.GetCurrentMacroPathName()

' Macro à exécuter à chaque reconstruction
stMacroMethods(0) = stFileName
'Module dans la macro
stMacroMethods(1) = "mdlProgramme"
' Procédure dans le module à exécuter
stMacroMethods(2) = «swmRegen»
```

On définit ici le point d'entrée correspondant à chaque reconstruction

```
' Macro à exécuter en édition
stMacroMethods(3) = stFileName
' Module dans la macro d'édition
stMacroMethods(4) = «mdlProgramme»
' Procédure dans la macro d'édition
stMacroMethods(5) = «swmEdition»
```

On définit ici le point d'entrée correspondant à une édition de la macro

```
' Facultatif, Macro à exécuter en cas de suppression ou effacement
stMacroMethods(6) = «»
' Facultatif, Module à exécuter en cas de suppression ou effacement
stMacroMethods(7) = «»
' Facultatif, Procédure à exécuter en cas de suppression ou effacement
stMacroMethods(8) = «»
```

On définit ici le point d'entrée correspondant à une suppression ou effacement de la macro

```
'On transforme les string en variant
vrMacroMethods = stMacroMethods

' Option de la Macro Feature
' swMacroFeatureByDefault ou swMacroFeatureAlwaysAtEnd
lgOption = swMacroFeatureByDefault

Set swFeatureMgr = swdoc.FeatureManager

'On insère la macro feature dans le feature manager du document ouvert avec le nom et les paramètres définis au dessus.
Set swMacroFeature = swFeatureMgr.InsertMacroFeature("Intervalle de Tolérance", "", _
vrMacroMethods, stFctName, vrFctType, vrFctValue, Nothing, swMacroFeatureByDefault)
```

Mécanisme pour insérer la macro feature dans l'arbre de construction du document

```
'On va à la procédure d'édition pour entrer les paramètres de la macro
Call swmEdition(swapp, swdoc, swMacroFeature)

End Sub
```

Appel de la procédure d'édition pour entrer les paramètres de la macro

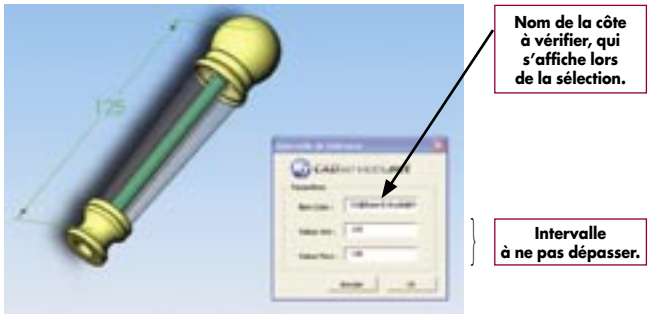
Edition de la Macro Feature :

Après avoir inséré la macro dans l'arbre avec la procédure «main» que nous venons de détailler ci-dessus, nous allons à présent détailler les fonctionnalités de chaque procédure qui sont appelées lorsque nous éditons la macro, reconstruisons le document ou supprimons cette macro.
Dans la procédure d'édition ci-dessous, nous allons tout d'abord récupérer les paramètres existants pour pouvoir les modifier en cas de rééditions, si c'est la première fois que l'on édite la macro feature, les paramètres sont vides.

```
'* Procédure qui s'exécute lors de l'édition de la macro feature

Function swmEdition(vrApp As Variant, vrPart As Variant, vrFeature
As Variant) As Variant
Dim MyFeature        As SldWorks.Feature
Dim swapp            As SldWorks.SldWorks
Dim swdoc            As SldWorks.ModelDoc2

'Récupération de la macro feature et accès aux données
Set MyFeature = vrFeature
Set swapp = vrApp
Set swdoc = vrPart

Set swMFdata = MyFeature.GetDefinition
'On récupère les paramètres de la macro Feature à restituer
swMFdata.GetParameters VarparamNames, VarparamTypes,
VarparamValues
'On appelle la procédure frmMain.SetEdition
Call frmMain.SetEdition(swapp, swdoc, vrFeature)

End Function
```

Une fois les paramètres récupérés on appelle alors la procédure «frmMain.SetEdition»
On utilise cette procédure «frmMain.SetEdition» pour afficher la boite de dialogue et entrer les paramètres qui vont permettre d'avoir des valeurs de côtes mini et maxi.

Boite de dialogue :



Reconstruction de la Macro Feature :

Une fois les paramètres renseignés et validés en cliquant sur «Ok», on exécute la procédure «swmRegen» du module «mldProgramme», elle sera re-exécutée ensuite pour chaque reconstruction.
Cette procédure permet de vérifier que la côte choisie est comprise dans l'intervalle de tolérance choisi :

```
'* Module qui est exécuté lorsque l'on fait une reconstruction ou
que l'on modifie les paramètres de la macro

Function swmRegen(vrApp As Variant, vrPart As Variant, vrFea-
ture As Variant) As Variant

  Dim MyFeature        As SldWorks.Feature
  Dim swDimention      As SldWorks.Dimension
  Dim dbValMin         As Double
  Dim dbValMax         As Double
  Dim dbDimention      As Double

  'Récupération des objets SW
  Set swapp = vrApp
  Set swdoc = vrPart
  Set swMacroFeature = vrFeature

  Set swMFdata = swMacroFeature.GetDefinition
  'On récupère les paramètres insérés par l'utilisateur
  swMFdata.GetParameters VarparamNames, VarparamTypes,
  VarparamValues

  'Si il n'y a pas de nom stocké alors on sort de la fonction
  If IsEmpty(VarparamValues) Then Exit Function

  'On récupère la cote
  Set swDimention = swdoc.Parameter(VarparamValues(0))

  'S'il n'y a pas de cote récupérée dans la frmMain alors on sort...
  If swDimention Is Nothing Then Exit Function

  'On transforme les intervalles de tolérance en double
  dbValMin = CDbl(VarparamValues(1))
  dbValMax = CDbl(VarparamValues(2))

  'On récupère la longueur de la cote avec les unités du système
  dbDimention = swDimention.GetSystemValue3(swThisConfigu
  ration, «»)(0) * 1000

  'Si la cote est trop petite
  If dbDimention < dbValMin Then
  'On affiche un message pour en informer l'utilisateur
     swapp.SendMsgToUser2 «La valeur de la cote
     « & CStr(VarparamValues(0)) & _
     « est trop petite», swMbWarning, swMbOk

  End If
```

```
'Si la cote est trop grande
If dbDimention > dbValMax Then
'On affiche un message pour en informer l'utilisateur
    swapp.SendMsgToUser2 «La valeur de la cote" &
    CStr(VarparamValues(0)) & _
    « est trop grande», swMbWarning, swMbOk

  End If

End Function
```

Si la valeur de la côte sélectionnée sort de l'intervalle, la macro nous affichera le message suivant :

Suppression de la Macro Feature :

Dans certain cas il est possible de spécifier une procédure dans le cas de suppression de la macro feature, dans notre exemple nous ne spécifions pas de procédure à exécuter.

```
' On spécifie le nom de la macro à exécuter, le nom du module et le nom
de la procédure que l'on exécute lors de la suppression

' Facultatif, Macro à exécuter en cas de suppression ou effacement
stMacroMethods(6) = «»
' Facultatif, Module à exécuter en cas de suppression ou effacement
stMacroMethods(7) = «»
' Facultatif, Procédure à exécuter en cas de suppression ou effacement
stMacroMethods(8) = «»
```

Ce qui signifie que lorsque que cette macro feature sera supprimée, aucun code ne sera exécuté.

Pour finir :

Quelques utilisations de macro feature :

– Alerte en cas de côtes, volumes ou masses dépassant un intervalle défini.

– Calcul automatique de propriétés (en fonction de dimensions pour générer un code article par exemple)

– Programmation de fonctions non prévues dans SolidWorks (hélice à pas variable, etc.)

– Dialogue avec une application externe (écriture ou lecture de dimensions dans un tableau Excel, etc.)

– Etc.

L'utilisation de macro feature dans vos documents peut vous apportez un grand nombre d'automatisme donc un gain de temps utilisateur.
Si vous percevez un intérêt pour votre entreprise dans l'utilisation de macro feature, sachez qu'Axemble propose des formations API qui peuvent vous familiariser avec ces mécanismes.

TIPS fourni par AXEMBLE
Pour en savoir plus : **www.mycadservices.net**