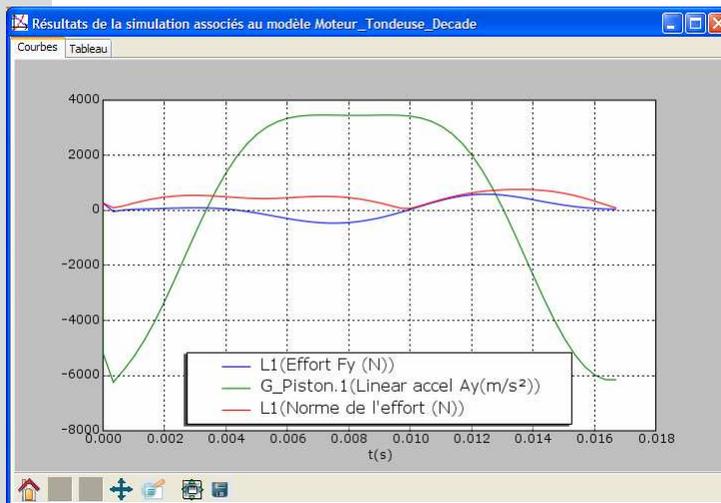
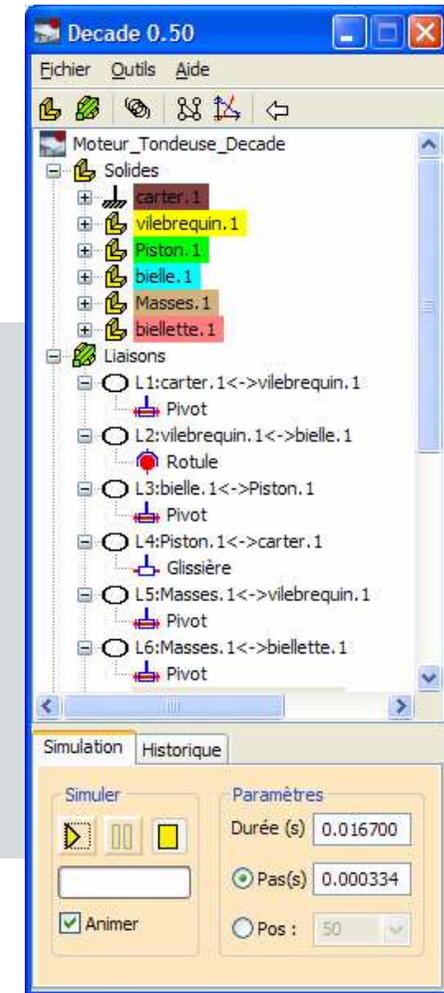




CATIA V5 Automation HybridShapeFactory Decade et autres exemples



Thomas Paviot
Lycée Dorian (Paris XI^{ème})
thomas.paviot@free.fr
thomas.paviot@decade-dynamics.org
www.decade-dynamics.org



De Decade vers l'HybridShapeFactory...

- Objectif : tracé de trajectoires sur l'ouvre-barrière Sinusmatic (maquette de Patrick Gorju – lycée Dorian) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShape Factory avec Python.
- Perspectives - Conclusion
- Questions / Réponses



BARRIERE SINUSMATIC

- E0 (E0.1)
- E1 (E1.1)
- E2 (E2.1)
- E3 (E3.1)
- E4 (E4.1)
- Paramètres
- Relations
- Contraintes
- Applications

Decade 0.53

- Solides
 - E0.1
 - E1.1
 - E2.1
- Repère
 - Points liés
 - O_E2.1
 - G_E2.1
 - Vecteurs
 - E3.1
 - E4.1
- Liaisons
 - L1: E0.1 <-> E1.1 (Pivot)
 - L2: E1.1 <-> E2.1 (Rotule)
 - L3: E2.1 <-> E3.1 (Glissière)
 - L4: E3.1 <-> E4.1 (Pivot)

Simulation Historique

Simuler Paramètres

Durée (s) 1.000000

Pas (s) 0.005000

Animer Pos : 200

Arête/Spline. 1/Trajectoire du point G_E2.1 sélectionné(e)

démarrer opensource@KMRC... Téléchargements fr.comp.lang.pytho... Microsoft PowerPoin... CATIA V5 - [BARRIE... C:\WINDOWS\sysste... Decade 0.53 FR 08:51

Trajectoire du centre de la liaison rotule



Problème technique

Comment piloter les ateliers surfaciques (en particulier le GSD) pour tracer dans une « Part » une trajectoire interpolée par une spline?



- Objectif : tracé de trajectoires sur le Sinusmatic (P.Gorju) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShape Factory avec Python : exemple d'une hélice
- Perspectives - Conclusion
- Questions / Réponses



Historique

- 1999 : idée/envie de réalisation d'un outil logiciel de simulation cinématique
- 2000 : naissance de pyMotion (enregistré chez SourceForge.net le 16 dec. 2001)(experimental)
- 23 sept. 2005 : naissance de Decade (enregistré chez gna.org) = pyMotion + simulation dynamique + Interface CAO
- 12 oct. 2005 : première version Decade 0.1
- 30 nov.2005 : première version publique Decade 0.20
- Janvier 2006 : dépôt du domaine decade-dynamics.org. Création d'un site communautaire www.decade-dynamics.org



Quelques constats

- Les logiciels de simulation dynamique du marché sont nombreux (Motion, Cosmos, Meca 3D, SIM Designer etc.) et différents dans leurs fonctionnalités et leur approche.
- Ils présentent des fonctionnalités avancées mais sont peu adaptés à une utilisation avec les élèves.
- Un constat d'amertume : la mésaventure Solid-Dynamics (été 2005)



Quel positionnement crédible pour un logiciel de simulation supplémentaire?

Ma réponse : un outil simple, transversal et libre.

Un élève de BTS doit être capable de conduire seul une étude cinématique/dynamique simple.

Le logiciel peut communiquer avec tous les logiciels de CAO du marché.
L'interface de Decade ne change pas.

L'accès au code source, ainsi que la possibilité de le modifier sont garantis par la licence GPL



Fonctionnalités actuelles (v0.50)

- Interfaçage CATIA V5(jusqu'à V5R15) et SolidWorks (de 2003->2006).
- Modélisation cinématique à partir de 6 liaisons élémentaires
- Possibilité de modification du modèle en cours de simulation
- Caractérisation des liaisons à partir des contraintes d'assemblage
- Animation dans Catia/SolidWorks
- Tracé de trajectoires (Catia V5)
- Affichage du graphe des liaisons
- Ressorts de traction/compression ($F=k.\Delta l$)
- Capteurs d'efforts, position, vitesse, accélération dans les assemblages ou pour des points liés
- Résultats sous forme de courbe/fichier CSV (pour un traitement avec Excel)
- Bilingue Français/Anglais
- ...



Quelques chiffres

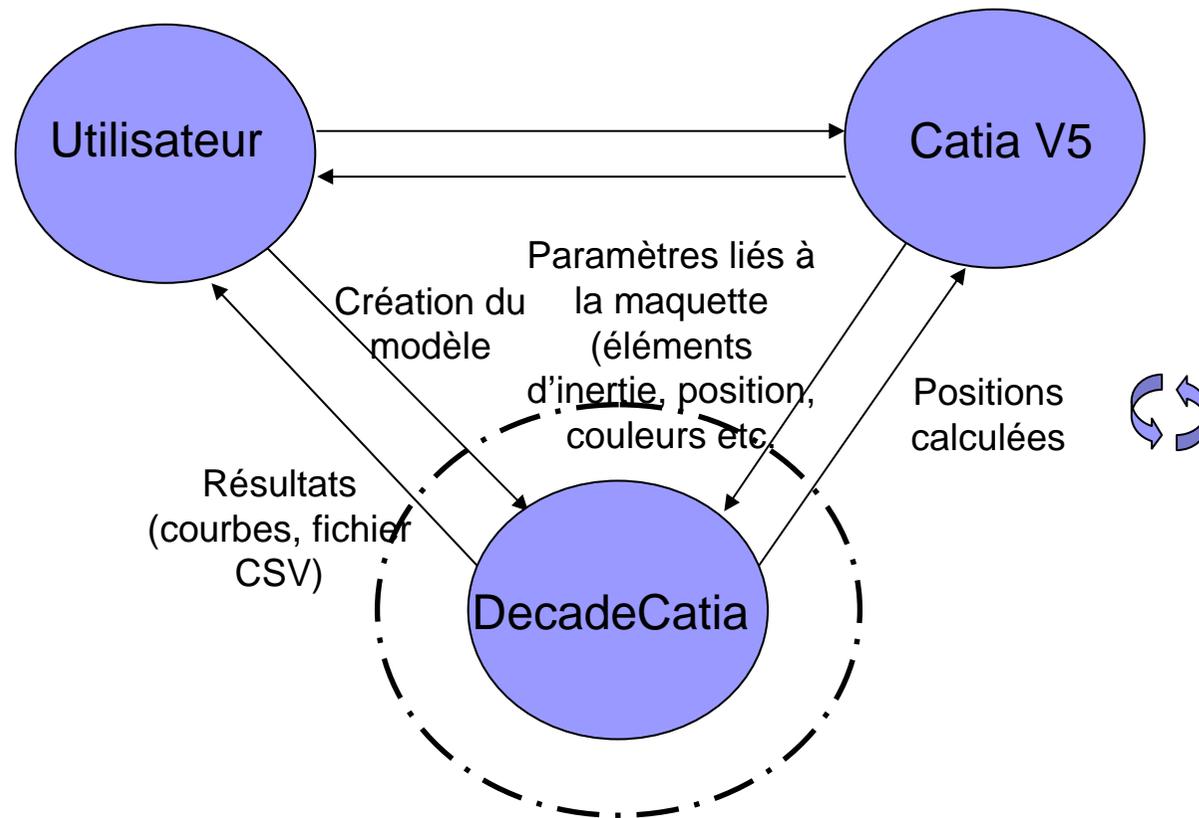
- 1^{ère} version publiée : Decade 0.20
- Version en cours : Decade 0.50
- Version à venir : Decade 0.53 (2006-sem.25)
- 1 release / 2 mois en moyenne
- 10000 lignes de code Python
- 624 téléchargements de Decade 0.50 (au 13 juin 2006=250 téléchargements/mois)
- 1500 visites/mois sur le site decade-dynamics
- 1 développeur – 1 chargé de la documentation (C.Faury)
- Une dizaine d'utilisateurs « impliqués »



- Objectif : tracé de trajectoires sur le Sinusmatic (P.Gorju) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShapeFactory avec Python : exemple d'une hélice.
- Perspectives - Conclusion
- Questions / Réponses

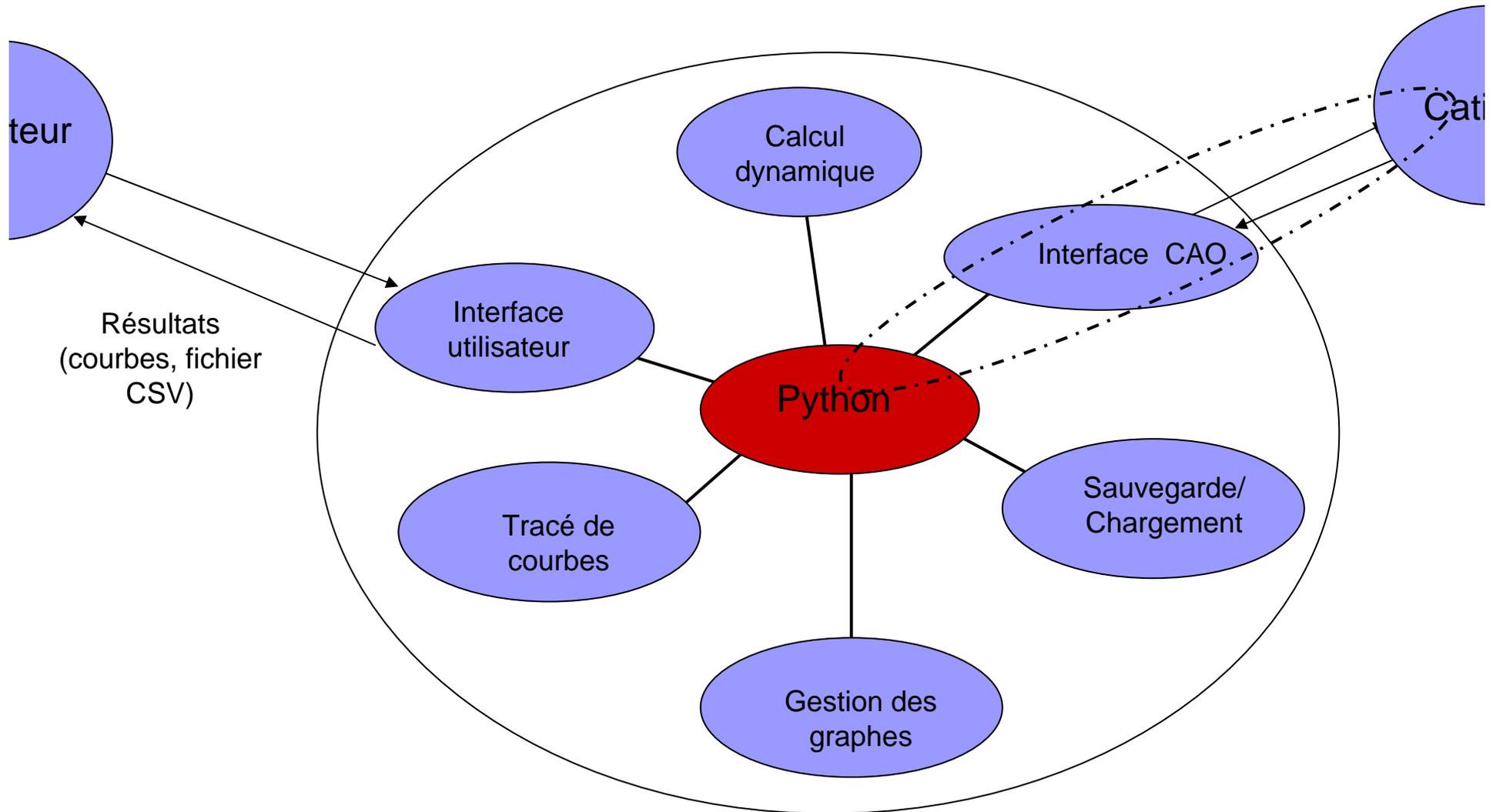


Decade : un logiciel indépendant communiquant





Decade : un Lego informatique





Le choix de Python : un pari réussi

Un langage de script orienté objet

- Possibilité d'exécution interprétée
- Communauté nombreuse et active
- Grand nombre de bibliothèques libres
- Développement rapide, code lisible
- Langage faiblement typé
- Pas de souci lié à la gestion du matériel



La communication Python/Catia

- Le pilotage de Catia par un programme extérieur (*via* Catia V5 Automation) est prévue pour Visual Basic
- Il est possible néanmoins d'utiliser Python, en se connectant au serveur COM installé par Catia V5 (en utilisant une bibliothèque Python spécialisée dans la gestion du système d'exploitation Windows).



- Objectif : tracé de trajectoires sur le Sinusmatic (P.Gorju) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShapeFactory avec Python : exemple d'une hélice.
- Perspectives - Conclusion
- Questions / Réponses



HybridShapeFactory

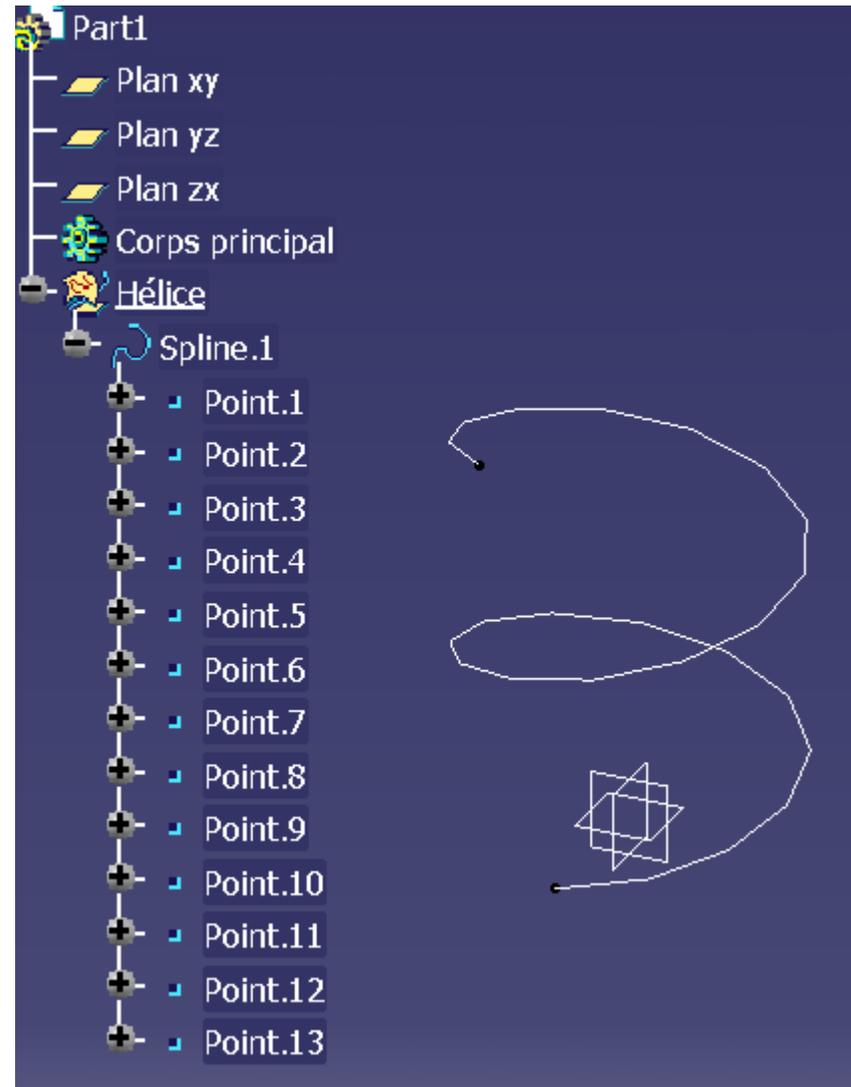
- ***Interface to create all kinds of HybridShape objects that may be needed in wireframe and surface design.***

Note:

This interface concern GSD/GSO/DL1 feature creation



Création d'une hélice avec HybridShapeFactory et Python





Exemple : Hélice (1/3)

■ Etape 1 : connexion à Catia

```
import win32com.client  
catia=win32com.client.Dispatch('Catia.application')
```

■ Etape 2 : création d'une nouvelle pièce

```
document=catia.Documents.Add("Part")
```



Exemple : Hélice (2/3)

- Etape 3 : création d'un set géométrique nommé « Hélice »

```
sets_geometriques =part.HybridBodies  
mon_set_geom = sets_geometriques.Add()  
mon_set_geom.Name="Hélice"
```

- Etape 4 : chargement du HybridShapeFactory et création d'une spline

```
shape_factory=part.HybridShapeFactory  
spline=shape_factory.AddNewSpline()
```



Exemple : Hélice (3/3)

- Etape 5 : création de points de coordonnées (x,y,z)

```
point=shape_factory.AddNewPointCoord(x,y,z)  
spline.AddPoint(point)
```

- Etape 6 : ajout de la spline dans le set géométrique

```
mon_set_geom.AppendHybridShape(spline)
```



- Objectif : tracé de trajectoires sur le Sinusmatic (P.Gorju) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShapeFactory avec Python : exemple d'une hélice.
- Perspectives - Conclusion
- Questions / Réponses



Conclusion

- Python + COM + Catia V5 offrent un potentiel infini.
- Le partage de la connaissance produit de la connaissance (partage≠échange)



Perspectives

- Rentrée 2006 : utilisation de Decade en BTS CPI à Dorian
- Travail qui reste à réaliser: ajouter les liaisons manquantes, possibilité de placer un effort sur une pièce ou dans un assemblage, gestion dynamique des collisions / contacts non permanents, augmenter le nombre d'interfaces, surfaces axoïdes, etc.
- En projet : un outil de gestion de coffre-fort numérique.



- Objectif : tracé de trajectoires sur le Sinusmatic (P.Gorju) à l'aide de Decade
- De 1999 à nos jours...
- Voyage au centre de Decade...
- Technique du pilotage de l'HybridShapeFactory avec Python : exemple d'une hélice.
- Perspectives - Conclusion
- Questions / Réponses



Annexe A

De quoi ai-je besoin pour exécuter le script fourni en exemple?

- Installer Python (en version 2004) : disponible sur le site <http://www.python.org>
- Installer ensuite « Python for Windows Extensions » : disponible sur le site <http://starship.python.net/crew/mhammond>

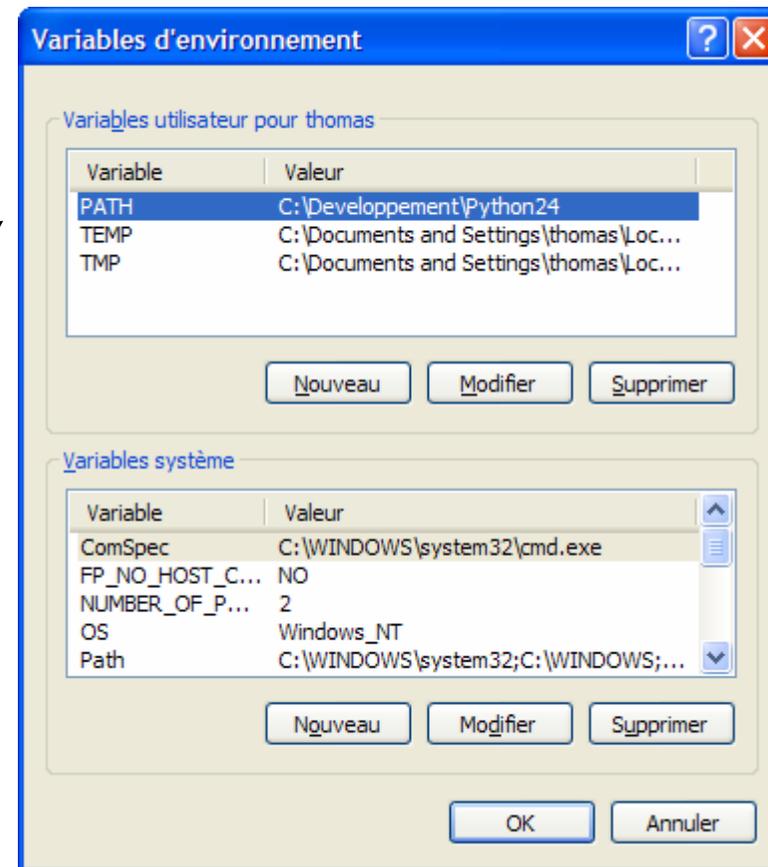
Lien direct vers la page de téléchargement de PyWin32:

http://sourceforge.net/project/showfiles.php?group_id=78018&package_id=79063&release_id=403008



Important : à la fin de l'installation de Python, vous devrez modifier la variable d'environnement PATH pour qu'elle pointe vers le répertoire d'installation de Python (vous pouvez aussi créer une variable PATH pour l'utilisateur en cours)

Fenêtre accessible depuis « Poste de travail », puis clic droit, « Propriétés ». Choisir alors l'onglet « Avancé » et cliquer sur le bouton « Variables d'environnement ».





Annexe B

Comment exécuter le script fourni en annexe?

- Ouvrir une fenêtre « DOS »: démarrer->Exécuter puis taper « cmd »



- Dans la fenêtre ainsi ouverte, se placer dans le répertoire ou se situe le script puis taper :

`python helice.py`

NOTE : il est également possible de double-cliquer sur l'icône « helice.py » pour lancer le script depuis un dossier Windows





Annexe C

Comment connaître les fonctions accessibles?

- Les objets, méthodes, propriétés de Catia V5 accessibles depuis Python sont les mêmes que celles accessibles depuis VB.
- La liste, ainsi que le documentation associée, de toutes ces fonctions sont disponibles dans le fichier « V5Automation.chm » situé dans le répertoire « intel_a/code/bin » de CatiaV5



Annexe D

Ça marche pas! Que faire?

- Il est possible que la connexion ne puisse pas être établie : Catia n'a peut être pas inscrit le serveur COM au moment de l'installation. Ceci peut être fait manuellement avec la commande:

`cnext.exe /regserver`

- Ne pas hésiter à m'envoyer un mail (coordonnées en page de garde)



Annexe D

Le script complet

Enregistrer le script suivant dans un fichier « helice.py ». La couleur rouge distingue les commentaires du reste du code.

```
# -*- coding: ISO-8859-1 -*-  
  
#  
# Tracé d'une hélice sous Catia V5 par interpolation d'un ensemble de points  
# Création d'une courbe 3D  
#  
  
#  
# Chargement du module permettant la connexion à un serveur COM  
#  
import win32com.client  
#  
# Chargement des fonctions sinus et cosinus depuis le module 'math'  
#  
from math import sin, cos  
  
#  
# Connexion au serveur COM  
#  
catia=win32com.client.Dispatch('catia.application') #->la communication est établie  
  
#  
# Création d'une nouvelle Part  
#  
document=catia.Documents.Add("Part")
```



```
#
# Part concernée
#
part=document.Part
#
# Création d'un nouveau set géométrique dans la part
#
sets_geometriques =part.HybridBodies
mon_set_geom = sets_geometriques.Add()
mon_set_geom.Name="Hélice"
#
# Chargement du hybrid_shape factory pour la pièce concernée
#
shape_factory=part.HybridShapeFactory
#
# Création de la spline
#
spline=shape_factory.AddNewSpline()
#
# Création des points de contrôle en coordonnées cylindro-polaires
#
values=[0,1,2,3,4,5,6,7,8,9,10,11,12]
for t in values:
    x=5*cos(t)
    y=5*sin(t)
    z=t
    # Création d'un point de coordonnées (x,y,z) dans le set géométrique
    point=shape_factory.AddNewPointCoord(x,y,z)
    #myHBody.AppendHybridShape(point)
    spline.AddPoint(point)

#ajout de la spline dans le set_géométrique
mon_set_geom.AppendHybridShape(spline)
#
# Mise à jour de la part
#
part.Update
```