Opérateur DETRUIRE

1 But

Détruire des concepts utilisateurs.

Après destruction, le concept ne peut donc naturellement plus être utilisé dans les commandes suivantes. La destruction d'un objet n'est pas si naturelle que cela puisse paraître. Il est important de bien comprendre son fonctionnement décrit dans cette documentation.

La destruction d'objets inutilisés permet de réduire l'encombrement des fichiers associés à la base 'GLOBALE'. Le mécanisme de retassage est pris en charge par le gestionnaire de mémoire au cours du travail. Cependant, un autre mécanisme de retassage peut être déclenché par l'utilisateur à l'aide du mot clé RETASSAGE = 'OUI' au sein de la procédure FIN [U4.11.02].

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
DETRUIRE      (
                  ♦  NOM           = lco,           [l_co]

                  ◊  INFO =     / 1,
                                / 2,

        )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérandes

3.1 Mot clé NOM

♦ NOM = lco

Liste des objets à détruire. On supprime toutes les références directes aux objets fournis dans le contexte où DETRUIRE est invoqué. Voir ci-dessous le fonctionnement détaillé.

3.2 Opérande INFO

◊ INFO = info

Dans le cas où INFO=2, la liste des objets détruits est imprimée dans le fichier MESSAGE.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Comment fonctionne la destruction ?

4.1 Exemple

On crée une liste de réels de nom `listr`
```
listr = DEFI_LIST_REEL( … )
```

On détruit le concept de nom `listr`
```
DETRUIRE(NOM=listr)
```

Notons que réutiliser la même variable Python `listr` a le même effet que `DETRUIRE` .

4.2 En détail

Les objets créés par code_aster (un maillage, un résultat) sont des objets Python.
`DETRUIRE` supprime les références à l'objet dans le jeu de commandes. Python supprime réellement cet objet quand il n'y a plus de référence vers cet objet.

Prenons l'exemple :
```
mesh = LIRE_MAILLAGE()
alias = mesh
lst = [mesh]
```

On a créé un objet *maillage* , `mesh` est une variable qui référence (ou « qui donne accès à », ou encore « qui pointe vers ») cet objet. `alias` est une variable qui référence le même objet. `lst` est une liste, créée avec l'opérateur [ ], qui contient un seul élément, et cet élément référence lui aussi le même objet maillage.

Si on fait :
```
DETRUIRE(NOM=mesh)
```

Toutes les références directes vers l'objet maillage sont supprimées du jeu de commandes.
Ainsi, les variables `mesh` et `alias` sont supprimées.
D'ailleurs, lors de l'exécution, on voit :
```
Suppression de la référence : 'mesh'
Suppression de la référence : 'alias'
```

En revanche, la référence via la liste n'est pas vue (il faudrait parcourir tous les types d'objets !) et donc, conformément au fonctionnement de Python, l'objet maillage n'est pas effectivement détruit puisqu'il existe encore une référence vers cet objet.

Si on supprime la liste :
```
del lst
```
ou l'élément de la liste :
```
del lst[0]
```
Alors il n'y a plus de référence vers l'objet qui est alors effectivement détruit. On peut voir :
```
Deleting 00000004 <MAILLAGE_SDASTER> mesh
```

Notons que :
```
DETRUIRE(NOM= lst[0] )
```
aurait eu exactement le même comportement (suppression de `mesh` et `alias` mais pas de `lst[0]`) car on passe l'objet à `DETRUIRE` et non la liste `lst` ou l'indice 0.

On aurait aussi le même comportement avec un dictionnaire contenant un objet maillage ou n'importe quel objet référençant le maillage.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

En effet, il n'y a que le contenant (l'objet liste, dictionnaire ou autre) qui sait comment supprimer un de ses éléments.

4.3 Dépendance entre objets

Les structures de données (champ aux nœuds-numérotation, champ par élément-modèle, etc …) s'appuient les unes sur les autres. DETRUIRE une structure de données qui est utilisée par une autre n'a donc aucun effet immédiat. En revanche, quand toutes les références à une structure de données sont supprimées, celle-ci est alors effectivement supprimée.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)