

Manuel de Descriptif Informatique
Fascicule D2.03 : Etablissement des catalogues
Document D2.03.01

Règles concernant l'écriture des Catalogues

Résumé :

Nous donnons dans ce document, les règles (ou conseils) que doit respecter le développeur lorsqu'il ajoute ou modifie un catalogue de commande ou un catalogue d'élément fini.

Table des matières

1 Présentation	3
2 Catalogues de commandes.....	3
3 Catalogues d'éléments.....	3
3.1 catalogue de PHENOMENE_MODELISATION.....	3
3.2 catalogue des GRANDEURS.....	3
3.3 catalogues des OPTIONS	4
3.4 catalogues des TYPE_ELEM.....	4

1 Présentation

Nous indiquons ici les règles (et conseils) concernant l'écriture :

- des catalogues de commandes [D5.01.01 §2]
- des catalogues d'éléments [D3.02.01 §3]

2 Catalogues de commandes

- 1) Utiliser le plus possible les possibilités du superviseur concernant les exclusions, les valeurs par défaut, ...
- 2) Ne pas utiliser la notion d'argument multiple.
- 3) Lorsqu'un argument est de type "texte" :
 - s'il peut prendre un nombre fini de valeurs, le définir en TXM et donner la liste intégrale des possibilités par le mot clé IN.
 - si c'est un texte libre : commentaire, ... utiliser TX
- 4) Les commentaires sont les bienvenus.
- 5) faire valider le vocabulaire par le Chef de Projet.

3 Catalogues d'éléments

3.1 catalogue de PHENOMENE_MODELISATION

- 1) Les noms des phénomènes et des modélisations doivent être validés par le Chef de Projet car ils apparaissent à l'utilisateur.

3.2 catalogue des GRANDEURS

- 1) Donner un nom aux grandeurs de la forme XXXX_S où S peut valoir :
 - R : réel
 - C : complexe
 - F : fonction (K8)
- 2) Lorsqu'on ne veut pas créer une nouvelle grandeur trop particulière, utiliser les grandeurs "neutres" : NEUT_R ou NEUT_K24.
- 3) Quand on modifie le catalogue des GRANDEURS, penser à mettre à jour le document "description des grandeurs" [D4.04.02] et classer les noms de grandeurs par ordre alphabétique.
- 4) Ne pas définir de grandeurs de type : L, K32, K80
- 5) Ne pas détruire de CMPS dans une grandeur existante sans avoir vérifié qu'aucun type_elem ne l'utilise.
- 6) Ne pas changer l'ordre des CMPS d'une grandeur existante sans modifier les type_elem qui l'utilisent.
- 7) Quand on introduit une nouvelle composante dans une grandeur, la mettre à la suite des CMPS existantes. Ceci évite de "casser" de la programmation trop "en dur" ; par exemple, un programmeur peut avoir fait :
 - vérification que 'DX' et 'DY' sont les deux premiers CMPS de la grandeur 'DEPL_R',
 - puis utilisation de DEPL_R(I), I = 1,2.

3.3 catalogues des OPTIONS

- 1) Les noms d'OPTIONS doivent être validés par le Chef de Projet s'ils apparaissent à l'utilisateur.
- 2) Ne pas réinventer les noms des paramètres pour chaque option ; s'inspirer de ceux déjà choisis. La forme usuelle est la suivante : `nom_par = 'P'//nom_gd`. Exemples : PMATERF, PGEOMER
- 3) Commenter chaque champ paramètre : exemple : PGEOMER, GEOM_R % champ de géométrie
- 4) Quand on ajoute une nouvelle option, penser à mettre des "-1" dans les catalogues de `type_elem` qui ne savent pas encore calculer cette nouvelle OPTION.

3.4 catalogues des TYPE_ELEM

- 1) Pour les noms des modes locaux s'inspirer des noms choisis par les `type_elem` voisins. Respecter l'usage :
 - Cxxxxx : mode de type carte
 - Nxxxxx : mode de type chamno
 - Exxxxx : mode de type chamelem
- 2) Pour un mode local de type CHAMELEM de type "N nno ...", s'assurer que nno est bien le nombre de noeuds du `type_maille` associé.
- 3) Quand on ajoute un nouveau `type_elem`, penser à mettre des "-1" en regard des OPTIONS existantes que le nouveau `type_elem` ne sait pas encore calculer.
- 4) Se poser des questions sur la cohérence du `type_elem` que l'on modifie avec les autres `type_elem` :
 - pourquoi le nouveau `type_elem` aurait-il un mode local "aux noeuds" alors que tous les autres l'ont "aux points de GAUSS" ?
 - pourquoi le nouveau `type_elem` n'utilise-t-il pas ce champ paramètre, cette CMP ?
 - ...