
Règles concernant l'écriture des catalogues

Résumé :

Nous donnons dans ce document, quelques 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 <code>type_element</code>.....	4

1 Catalogues de commandes

- Utiliser le plus possible les possibilités du superviseur concernant les exclusions, les valeurs par défaut, ...
- 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
- Les commentaires sont les bienvenus.
- Faire valider le vocabulaire en RTA.

2 Catalogues d'éléments

2.1 Catalogue de PHENOMENE_MODELISATION

- Les noms des phénomènes et des modélisations doivent être validés par l'EDA / RTA car ils apparaissent à l'utilisateur.

2.2 Catalogue des GRANDEURS

- Donner un nom aux grandeurs de la forme `XXXX_S` où S peut valoir :
 - R : réel
 - C : complexe
 - F : fonction (K8)
- Lorsqu'on ne veut pas créer une nouvelle grandeur trop particulière, utiliser les grandeurs "neutres" : `NEUT_R` ou `NEUT_K24`.
- 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.
- Ne pas définir de grandeurs de type : L, K32, K80
- Ne pas détruire de CMPS dans une grandeur existante sans avoir vérifié qu'aucun `type_element` ne l'utilise.
- Ne pas changer l'ordre des CMPS d'une grandeur existante sans modifier les `type_element` qui l'utilisent.
- 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.

2.3 Catalogues des OPTIONS

- Les noms d' OPTIONS doivent être validés par l'EDA / RTA s'ils apparaissent à l'utilisateur.
- 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
- Commenter chaque champ paramètre : exemple : PGEOMER, GEOM_R << champ de géométrie. Provient du maillage. >>
- Quand on ajoute une nouvelle option, penser à mettre des "-1" dans les catalogues de `type_element` qui ne savent pas encore calculer cette nouvelle OPTION.

2.4 Catalogues des `type_element`

- Pour les noms des modes locaux s'inspirer des noms choisis par les `type_element` voisins. Respecter l'usage :
 - Cxxxx : mode de type carte
 - Nxxxx : mode de type `cham_no`
 - Exxxx : mode de type `cham_elem`
- Quand on ajoute un nouveau `type_element`, penser à mettre des "-1" en regard des OPTIONS existantes que le nouveau `type_element` ne sait pas encore calculer ; le code s'arrête en erreur fatale avec "-1".
- Se poser des questions sur la cohérence du `type_element` que l'on modifie avec les autres `type_element` :
 - pourquoi le nouveau `type_element` aurait-il un mode local "aux noeuds" alors que tous les autres l'ont "aux points de GAUSS" ?
 - pourquoi le nouveau `type_element` n'utilise-t-il pas ce champ paramètre, cette CMP ?
 - ...