Procedure IMPR_CONCEPT

1 Goal

To print in a displayable form fields of “given” starting from concepts affected by the user with the orders AFFE_MATERIAU , AFFE_CARA_ELEM and AFFE_CHAR_MECA .

2 Syntax

```
IMPR_CONCEPT     (

◊   UNIT   =  links,
◊   FORMAT =  /   'MED',                                    [DEFECT]
              /  'RESULT',
◊   INFORMATION =    / 1,                                         [DEFECT]
              / 2,
If format = 'MED':
{
   ◊   VERSION_MED   =  /   '3.3.1',                        [DEFECT]
                        /   '4.0.0',
                        /   '4.1.0',
}

♦   CONCEPT    =   _F (
   /   CHAM_MATER    = chmat,                               [cham_mater]
   /   CARA_ELEM     = carele,                              [cara_elem]
      ◊   REPERE_LOCAL          =   / 'NOT' ,               [DEFECT]
                                    / 'ELEM',
                                    / 'ELNO',
      # if REPERE_LOCAL = 'ELEM' or 'ELNO',
            ♦      MODEL     = Mo                           [model]
   /   LOAD          = charg,                               [load]
      )
)
```

3 Typography in documentation IMPR_CONCEPT

Procedure IMPR_CONCEPT allows to print in a displayable form the quantities affected by the user with the orders AFFE_MATERIAU , AFFE_CARA_ELEM and AFFE_CHAR_MECA .
That makes it possible to check that the rules of overload of the code lead to the assignments wished.

For example:

```
IMPR_CONCEPT(FORMAT=' MED',
              CONCEPT= (
                       _F (CHAM_MATER = CHAMPMAT),
                       _F (CARA_ELEM  = CARA_ELE),
                       _F (LOAD      = CHARG1),),
             )
```

The fields contained in these structures of data are printed in two very different forms:

- The form “with” (easiest to interpret graphically): each component is separately printed like a real number. For example, the thickness of the elements of hull is displayable like a scalar field (scalar map in Salomé).
- The form “B”: one assigns to each mesh a whole code: 1,2,3,… N. The meshs affected by the same code have ALL then their identical components. The “definition” of the codes, i.e. the values of these components is printed in the file message. One can visualize the “codes” like a scalar field, which makes it possible “to see” the zones where “all is constant”.

Form “A” is used systematically for the format “MED”, except for the material field because this field contains the name of affected material on the meshs and this name is not a number.
Form “A” is not programmed with the format “RESULT”.

The form “B” is systematically used for the format “RESULT” and for the material field.

For the form “B”, the correspondence between the affected quantities and the code is given in the file .mess.

For example, for the material field:

```
IMPRESSION OF A FIELD OF CONCEPT: MATERIAL field
NAME OF THE FIELD: CHAMPMAT_CHAMP_MAT
CORRESPONDENCE VALUE <-> CONTAINED:
VALUE =          1.
    X1       = MAT_1
VALUE =          2.
    X1       = MAT_2
VALUE =          3.
    X1       = MAT_3
```

4 Operands FORMAT and UNIT and INFORMATION

4.1 Operand FORMAT

The operand FORMAT allows to specify the format of the file where to write the result.
Two formats 'MED' and 'RESULT' are available. LE format 'MED' is the format of writing by default.

4.2 Operand `UNIT`

Defines in which unit one writes the file med. By default, `UNIT` = 80 and corresponds to the unit by default of the type `rmed` in astk.

4.3 Operand `INFORMATION`

The keyword `INFORMATION` when it is equal to 2 makes it possible to obtain information on the impressions carried out by the order.

4.4 Operand `VERSION_MED`

```
◊  VERSION_MED  =  /'3.3.1',                    [DEFECT]
                   /'4.0.0',
                   /'4.1.0',
```

During the creation of a new file to the format med, the impression is made with the format med 3.3.1.
If the file already exists, the level of format med is preserved.
One can change the version of file MED with the keyword `VERSION_MED` .

5 Keyword factor `CONCEPT`

5.1 Operand `REPERE_LOCAL`

SI `REPERE_LOCAL` the value has '`ELEM`' or '`ELNO`', the keyword `MODEL` is obligatory.

If `REPERE_LOCAL` the value has '`ELEM`', the 3 vectors constituting the local reference mark of each element are printed, to be able to be visualized with L' helps of ParaVis.

If `REPERE_LOCAL` the value has '`ELNO`', the 3 vectors constituting the local reference mark of each element are recorded in the form of field with the nodes, to be able to be then used in the computer of ParaVis in combination with D' other fields with the nodes. In this case no other information coming from the concept `CARA_ELEM` is not recorded in the file.

When `REPERE_LOCAL=' ELNO'` it is thus possible to combine the local vectors with the components of the internal fields of efforts. That makes it possible to visualize the vectors efforts in 3D, like carrying out an animation of their evolution during moments of calculation. To carry out this action several elementary operations are to be realized in ParaVis:

- To open the file `MED` containing the reference marks:
  on `REPE` - > Filter "`ELNO Points`" - > Filter "Merge blocks"
- To open the file `MED` containing the fields:
  on `EFGE_ELNO` - > Filter "`ELNO Points`" - > Filter "Merge blocks"
- selection of the 2 "Merge blocks" then "Suspends Attributes"

In "Calculator" one has access to the vectors of `REPE` and with the components of the fields.
The vector effort (NR, Vy, Vz) is calculated in the following way:

Fint = CAREL___.REPLC_1∗xxxxxxxxEFGE_ELNO_N +
CAREL___.REPLC_2∗xxxxxxxxEFGE_ELNO_VY +
CAREL___.REPLC_3∗xxxxxxxxEFGE_ELNO_VZ

where 'xxxxxxxx' indicate the name of the concept result, produced by *Code_Aster*.

The vector Moment (MT, My, Mz) is calculated in the following way:

Mint = CAREL___.REPLC_1∗xxxxxxxxEFGE_ELNO_MT +
CAREL___.REPLC_2∗xxxxxxxxEFGE_ELNO_MFY +
CAREL___.REPLC_3∗xxxxxxxxEFGE_ELNO_MFZ

where ‘xxxxxxxx’ indicate the name of the concept result, produced by *Code_Aster*.

5.2 Fields being able to be visualized

CHAM_MATER:
Material field

CARA_ELEM:
General characteristics of the bars
Characteristics géom. bars
General characteristics of the beams
Characteristics géom. beams
Wire specifications
Characteristics of the curved beams
Characteristics of the “fluid” beams
Characteristics of the discrete elements K_*
Characteristics of the discrete elements M_*
Characteristics of the discrete elements A_*
Characteristics géom. hulls
Orientation of the elements 2D and 3D
Orientation of the hulls and the beams

LOAD :
Loading of GRAVITY
Loading of ROTATION
Loading of PRES_REP
Loading of voluminal forces in 3D
Loading of surface forces in 3D
Loading of linear forces in 3D
Loading of surface forces in 2D
Loading of linear forces in 2D
Loading of forces distributed for the hulls
Loading of PRE_EPSI
Loading of FORCE_ELEC
Loading of FLUX_THM_REP
Loading ofIMPE_FACE
Loading ofONDE_FLUI