Author: Marc Poinot, ONERA/DSNA
Contact: marc.poinot@onera.fr

Rigid motion applied to a Family for a set of Zones

The proposal is a modification of the *Family_t* node. The modification re-use existing CGNS structures, but it implies some constraints to *Zone_t* node and sub-nodes.

We propose to allow one *RigidGridMotion_t* nodes under the *Family_t* node. The application of the motion is distributed on each zone having a *FamilyName_t* *FamilyName* value corresponding to the *Family_t* node name.

```
Family_t :=
   {
   List( Descriptor_t Descriptor1 ... DescriptorN ) ;                (o)
   FamilyBC_t FamilyBC ;                                             (o)
   List( GeometryReference_t
         GeometryReference1 ... GeometryReferenceN ) ;               (o)
   RotatingCoordinates_t RotatingCoordinates ;                       (o)

   RigidGridMotion_t RigidGridMotion;                                          (o)

   List( UserDefinedData_t UserDefinedData1 ... UserDefinedDataN ) ; (o)
   int Ordinal ;                                                     (o)
   } ;
```

Constraints & Remarks:

1- The use of a set-of-zones structure usually located in the `Zone_t` node raises the issue of *Family* related data: the actual information is not in the *Zone* but elsewhere. As a matter of fact, in the case of a *Family* rigid motion, there is no way to indicate into the `Zone_t` the fact that it is a rigid motion, the application has to find if the *Zone* has a *Family* with motion data or not.

2- The use of a *Family* rigid motion in at least one *Family* of a `CGNSBase_t` implies no Motion node in all `Zone_t` of this `CGNSBase_t` (i.e. no rigid or arbitrary motion).

3- In the case of a rigid motion node declared at the `Family_t` level, no *RigidGridMotionPointers* should be found at the `Zone_t` iterative data level.

4- The `FamilyPointers` in the `BaseIterativeData_t` node and the use of multiple *Families* with different *RigidMotions* would be used to define a time-dependant *Family Rigid* motion.