

Construction de scène SOFA à partir du PML/LML

1 – Structure générale du PML « sofaisé »

On part du principe que PML n'est pas utilisé pour décrire l'ensemble des données de scène de SOFA.

Par contre un document PML "sofaisé" peut être chargé directement dans SOFA à l'aide de classes d'importation et de classes de représentation.

Le PML doit avoir au minimum la structure suivante :

```
<atoms>
index + position de tous les points du modèle
</atoms>
<exclusiveComponent>
  <multiComponent name="bodies">
    <structuralComponent bodyType="type de body" name="nom">
      topologie du body
    </structuralComponent> 1 structuralComponent par body
  </multiComponent>
</exclusiveComponent>
<informativeComponent>
autres informations non indispensables (affichage, habillage...)
</informativeComponent>
```

2 – Description des Body

La multiComponent exclusive "bodies" contient une structural component par corps/objet de la scène/modèle.

Chaque "body" a au minimum les attributs suivants :

- **nom** ="nom libre"
- **bodyType** ="rigid" ou bodyType="nomForceField" ou bodyType="Interaction"

2.1 – Rigides (bodyType="rigid")

Il s'agit d'un corps rigide indéformable. Cette SC contient soit :

- une cellule de type="POLY_VERTEX", contenant la liste des atomes
- nc cellules de types="TRIANGLE"/"QUAD", si on a une représentation surfacique

Un seul attribut est obligatoire, au choix :

- **mass** ="1 ou n valeurs" : les valeurs de masse attribuées aux n atoms constitutifs (1 valeur = masse uniforme)
- **inertiaMatrix** ="1, 3 ou 6 valeurs" : 1 valeur = matrice d'inertie isotrope (equiv. 1 sphère), 3 valeurs = matrice diagonale, 6 valeurs = matrice symétrique

Les attributs optionnels sont :

- **position** ="3 ou 6 valeurs" : c'est l'équivalent d'une translation/rotation initiale des atomes.
 - Par défaut position = 0
 - Position avec 3 valeurs = tx ty tz
 - Position avec 6 valeurs = tx ty tz rx ry rz
 - Position avec 7 valeurs = tx ty tz rx ry rz rw
- **velocity** ="3 ou 6 valeurs" : c'est la vitesse initiale de la translation/rotation
 - Par défaut velocity = 0
 - Velocity avec 3 valeurs = vtx vty vtz
 - Velocity avec 6 valeurs = vtx vty vtz vrx vry vrz
- **collision** ="booléen" : permet d'enclencher ou non le traitement des collisions pour ce corps.

Attention : si on a plusieurs corps rigides de la même forme (utilisant le même nombre d'atomes) il faudra dupliquer les atomes dans le PML...(on peut imaginer une optimisation de la taille du fichier en décidant que la duplication n'est pas obligatoire mais sera fait automatiquement dans PMLReader si plusieurs bodies de type "rigid" font appels aux mêmes atomes, mais pas dans un premier temps).

Remarque :

Le nombre total de ddl de SOFA n'est pas strictement égal au nb total d'atomes, les rigides étant mappés à partir d'un ddl référentiel (on a donc nb ddl = nb atoms +1).

2.2 – Forcefields

Il s'agit d'un corps rigide déformable. Deux types sont pour l'instant implémentés : FEM et StiffSpring.

Pour l'ensemble des forcefields, un seul attribut est obligatoire, au choix :

- **mass** ="1 ou n valeurs" : les valeurs de masse attribuées aux n atoms constitutifs (1 valeur = masse uniforme sur chaque atome).
- **density** ="1 ou m valeurs" : la densité (masse volumique pour être exact) de l'objet, la masse de chaque atome est alors calculée en fonction du volume "contrôlé".

Un seul attribut optionnel :

- **collision** ="booléen" : permet d'enclencher ou non le traitement des collisions pour ce corps.

Attention : En biomécanique classique l'organisation est basée sur la structure des différents objets/bodies/matériaux (par exemple : un body par organe). Dans SOFA, afin d'optimiser les calculs et de simplifier la représentation et les interactions, l'organisation est basée sur les différents degrés de liberté. Ainsi, si deux body forcefield de même type ont des ddl en communs, ceux-ci seront fusionnés dans Sofa pour ne faire qu'un seul body. Les masses des ddl en communs sont additionnées.

2.2.1 – FEM (bodyType="FEM")

Pour les objets de type éléments finis, la SC doit contenir nc cellules de type TETRAHEDRON ou HEXAHEDRON. Attention : Les hexaèdres seront divisés en 5 tétraèdres pour être compatible avec la topologie de Sofa.

Les attributs spécifiques (et optionnels) aux body FEM sont les suivants :

- **young** = valeur pour le module d'young, en Pascal (5000 par défaut)
- **poisson** = valeur pour le coef de poisson (0.45 par défaut).
- **deformation** = SMALL, LARGE ou POLAR, pour le type de déformation.

2.2.2 – StiffSprings (bodyType="stiffSpring")

Pour les objets de type masse-ressort, la SC doit contenir nc cellules de type définissant des arrêtes (line, triangle, quad, hexahedron ou tetrahedron). Les ressorts sont créés directement à partir de ces arrêtes.

Les attributs spécifiques (et optionnels) aux body StiffSprings sont les suivants :

- **stiffness** = valeur pour la raideur des ressorts, 500 par défaut.
- **damping** = valeur pour l'ammortissement, 5 par défaut.

2.3 – Interaction (bodyType="interaction")

L'interaction permet de lier deux body différents avec des ressorts. Ce body n'a pas de ddl qui lui est propre. Il est défini par des cellules de types LINE dont les extrémités appartiennent respectivement aux deux objets à lier.

Deux attributs sont obligatoires :

- **body1** = nom du premier objet à lier.
- **body2** = nom du second objet à lier.

Les attributs optionnels sont les suivants :

- **stiffness** = valeur pour la raideur des ressorts, 500 par défaut.
- **damping** = valeur pour l'ammortissement, 5 par défaut.

3 – les LML

Trois types de chargements/conditions limites sont pour l’instant gérés : les déplacements imposés, les déplacement nuls et les forces.

Les fichiers LML n’ont pas besoin d’être « sofaitsés », les changements d’indices entre atoms et dofs sont gérés automatiquement.

Attention : les contraintes (points fixes et déplacements imposés) ne peuvent s’appliquer qu’à un rigide dans son ensemble. L’explication est que le mapping impose la position des points du maillage à partir de la position du ddl référenciel, et non l’inverse. Par contre les contraintes sur les body forcefields ainsi que les forces peuvent s’appliquer sur n’importe quel point du maillage.

4 – Importation dans Sofa

L’importation d’une scène PML dans Sofa se fait avec la classe PMLReader. Il suffit de passer en paramètre le nom du fichier PML, ainsi que le noeud racine de la scène.

De la même façon, l’importation de LML se fait avec la classe LMLReader.

La construction de la scène se fait alors en appelant la méthode « buildStructure » de chacun de ces deux objets.

La classe PMLReader contient une factory où son enregistrés les classes de traductions de chaque type de body.

Il existe une classe PMLBody abstraite dont héritent les classes de traductions (PMLRigidBody, PMLFemForcefield, PMLStiffSpringForcefield et PMLInteractionForcefield).

Pour ajouter un nouveau type de body, il faut donc créer une nouvelle classe de traduction héritant de PMLBody, et l’enregistrer dans la factory de PMLReader.

Pour la traduction des LML, la classe LMLReader, utilise deux classes : LMLConstraint et LMLForce.

LMLConstraint (héritant de la classe Sofa Constraint) traduit les chargements de types translation et points fixes.

LMLForce (héritant de la classe Sofa ForceField) traduit les chargements de type force.