**Organisation du nouveau DServer XPS V2**

On va avoir une arborescence de devices Tango dans le Dserver XPSV2 :

**XPSGroup :** contrôle le groupe connu dans le XPS sous le nom MultipleAxes. On pourra étendre par la suite à d’autres groupes : par exemple à SingleAxisGroup pour permettre de définir un axe comme esclave d’1 autre. XYGroup ou XYZGroup pourraient un jour être utiles pour un porte-échantillon, par exemple. On interface les fonctions XPS de la forme Groupxxx, SingleAxisxxx, XYxxxx, XYZxxx,…

**XPSAxis** contrôle des fonctions d’axe connues dans le XPS sous les noms de la forme Positionerxxx

**XPSTrajectory** contrôle des fonctions de trajectoires : au départ uniquement les trajexctoires de type PVT, qu’on pourra étendre ensuite à des trajectoires PT, avec calcul dans le device des vitesses, ou même de P, avec calcul dans le device des temps et vitesses.  
De même si d’autres types de groupes étaient définis, on pourrait avoir besoin des trajectoires qui leurs sont associés : par exemple XYGroup et les trajectoires line-arc, XYZGroup et les trajectoires spline.

Dans la mesure où on a un device avec des attributs dynamiques, on est obligés d’avoir 1 seul device XPSGroup par instance : chaque groupe doit être démarré dans sa propre instance, avec ses petits XPSAxis.

Nous choisissons de localiser le code « utile » dans des classes C++ pures qui pourront échanger leurs données respectives sans limitations et qui permettront de factoriser l’accès au HW et à garantir que les données sont uniques dans le code : par exemple toutes les classes partagent l’offset qui est détenu dans la classe Positioner. Les devices Tango seront pour l’essentiel des coquilles autour des classes C++.

Ces devices Tango partagent des informations qui doivent être mises à jour simultanément :

* Des informations qui restent dans Tango : Offset
* Des informations communes : position, vitesse, consigne de position, status, state…

Certaines de ces informations doivent être maintenues persistantes dans Tango : on doit mémoriser l’offset, la consigne de vitesse courante, …  
Cela pose la question de savoir quelle stratégie permettra cette mémorisation :   
par exemple pour l’offset : la solution simple est de n’avoir accès en écriture à l’offset (la vitesse) que dans le device XPSAxis (et pas dans XPSGroup) pour bénéficier du mécanisme d’attribut mémorisé de Tango.

La solution permettant l’accès en écriture à l’offset, la vitesse,… simultanément dans XPSGroup et XPSAxis nécessiterait un gros effort de développement pour avoir la mémorisation de la valeur, sans avoir la garantie que ça marcherait dans des conditions limites.

Un autre point concerne les soft limits : Dans XPSAxis V1, elles étaient définies (et mémorisées) en utilisant les properties d’attribut min\_value, max\_value. On peut à la rigueur conserver le mécanisme dans le nouveau device XPSAxis V2, mais ça me semble très difficile de porter ce mécanisme pour des attributs dynamiques dans XPSGroup V2.  
Je propose donc que ces soft limits ne soient accessibles que via XPSAxis.

J’en viens à me demander si ce serait possible d’écrire/lire des variables utilisateur dans le XPS et de les mémoriser dans des fichiers, pour contourner le problème et simplifier le code du XPS V2.