Detector description language news
News of the geometry
  - No more need for default attributes
  - Transformations
    - Usage
    - Composition
  - Boolean solids
  - Parametrized volumes
  - Trapezoids

News of detector elements
  - userParameter and userParameterVector

Done / To do list
Null or default values are no more needed inside XML.

These are mostly:

- 0*m for every dimension
- 0*degree for every angle
- 360*degree for deltaPhiAngle
- 180*degree for deltaThetaAngle

**Old Way**

```
<tubs name="...">
  sizeZ = "1*m"
  innerRadius = "0*m"
  outerRadius = "1*m"
  startPhiAngle = "0*degree"
  deltaPhiAngle = "360*phi"/
</tubs>
<posXYZ x="0*m"
  y="1*m"
  z="0*m"/>
```

**New Way**

```
<tubs name="...">
  sizeZ = "1*m"
  outerRadius = "1*m"/
</tubs>
<posXYZ y="1*m"/>
```

**Old Way**
Usage of the transformations

- No more inside solids but outside
- Accepted after every solid (even first one in booleans)
- Give position first, rotation afterwards. Rotation is applied first.

**Old Way**

```xml
<subtraction name="L">
  <box .../>
  <box ...>
    <posXYZ .../>
  </box>
</subtraction>
```

**New Way**

```xml
<subtraction name="L">
  <box .../>
  <!-- I can put pos and rot here -->
  <box .../>
  <posXYZ .../>
</subtraction>
```
Composition of transformations

- A new tag: `<transformation>`
- No attributes, as many children as you want
- It just applies the transformation in the order they are given

```
<transformation>
  <pos .../>
  <rot .../>
  <transformation>
    <pos .../>
    <rot .../>
  </transformation>
  <pos ...>
  </transformation>
</transformation>
```
Nothing new but a recursive behavior of booleans.

```xml
<subtraction name="...">
  <union name="...">
    <tubs .../>
    <tubs .../>
    <posXYZ .../>
  </union>
  <intersection name="...">
    <tubs .../>
    <tubs .../>
    <posXYZ .../>
  </intersection>
</subtraction>
```
Nothing new but a recursive behavior of parametrization.

```xml
<logvol ... name="ppv">
  <paramphysvol number="4">
    <physvol ... logvol=".../vol"/>
    <posXYZ y="1*m"/>
    <rotXYZ rotZ="45*degree"/>
  </paramphysvol>
</logvol>

<logvol ... name="pppv">
  <paramphysvol number="3">
    <physvol ... logvol=".../ppv"/>
    <posXYZ/>
    <rotXYZ rotZ="45*degree"/>
  </paramphysvol>
</logvol>
```
2D and 3D parametrizations

- New tags: `<paramphysvol2D>` and `<paramphysvol3D>`
- Special behavior, different from composition
  - All rotations applied first
  - Positionning done at the end
- The results are "grids"

```xml
<logvol ... name="ppv2D">
  <paramphysvol2D number1="4"
  number2="4">
    <physvol ... logvol=".../vol"/>
    <posXYZ y="1*m"/><rotXYZ rotZ="45*degree"/>
    <posXYZ x="1*m"/><rotXYZ rotZ="90*degree"/>
  </paramphysvol2D>
</logvol>
```
Example of 3D parametrization

```xml
<logvol ... name="ppv3D">
  <paramphysvol3D number1="4"
                   number2="4"
                   number3="2">
    <physvol ... logvol=".../vol"/>
    <posXYZ y="1*m"/>
    <rotXYZ rotZ="45*degree"/>
    <posXYZ x="1*m"/>
    <rotXYZ rotZ="90*degree"/>
    <posXYZ x=".5*m" y=".5*m"/>
    <rotXYZ rotZ="180*degree"/>
  </paramphysvol3D>
</logvol>
```
General trapezoids

- New tag: `<trap>` with lots of attributes: `name`, `sizeZ`, `theta`, `phi`, `sizeY1`, `sizeX1`, `sizeX2`, `alp1`, `sizeY2`, `sizeX3`, `sizeX4`, `alp2`.
- This is exactly the **geant general trapezoid**.

```xml
<trap name="trap_sample"
    sizeZ="12*m"
    theta="0*degree"
    phi="0*degree"
    sizeY1="2*m"
    sizeX1="2*m"
    sizeX2="4*m"
    alp1="0*degree"
    sizeY2="4*m"
    sizeX3="4*m"
    sizeX4="8*m"
    alp2="0*degree"/>
```
Two new tags: `<userParameter>` and `<userParameterVector>`

- Attributes are `name`, `type` and `comment`. All these are strings.
- The value is given directly between opening and closing tags.
- It is accessible in regular interface `IdetectorElement` via methods `userParameterType`, `userParameterComment`, `userParameterValue`, `userParameter` (see next talk)

```xml
<userParameter
    name="Al_plate_thickness"
    type="double"
    comment="blabla">
  1.2222*mm
</userParameter>

<userParameterVector
    name="Al_plate_thickness"
    type="double"
    comment="blabla">
  1.222*m 1.333*m 1.444*m 1.555*m
</userParameterVector>
```
To be done / discussed

- Concerning geometry
  - Test general trapezoids
  - Discuss the memory problem for parametrized physical volumes
  - Be able to reload xml in GaudiLab (most of the work in DetDesc)
  - Usage of a transformation for the first solid in a boolean
  - Compile and test everything under windows
  - Problems with GaudiLab (stability, boolean operations, ...)

- Concerning structure
  - Discuss the scope of the `<parameter>` tag
  - Improve the `<userParam(Vector)>` tag