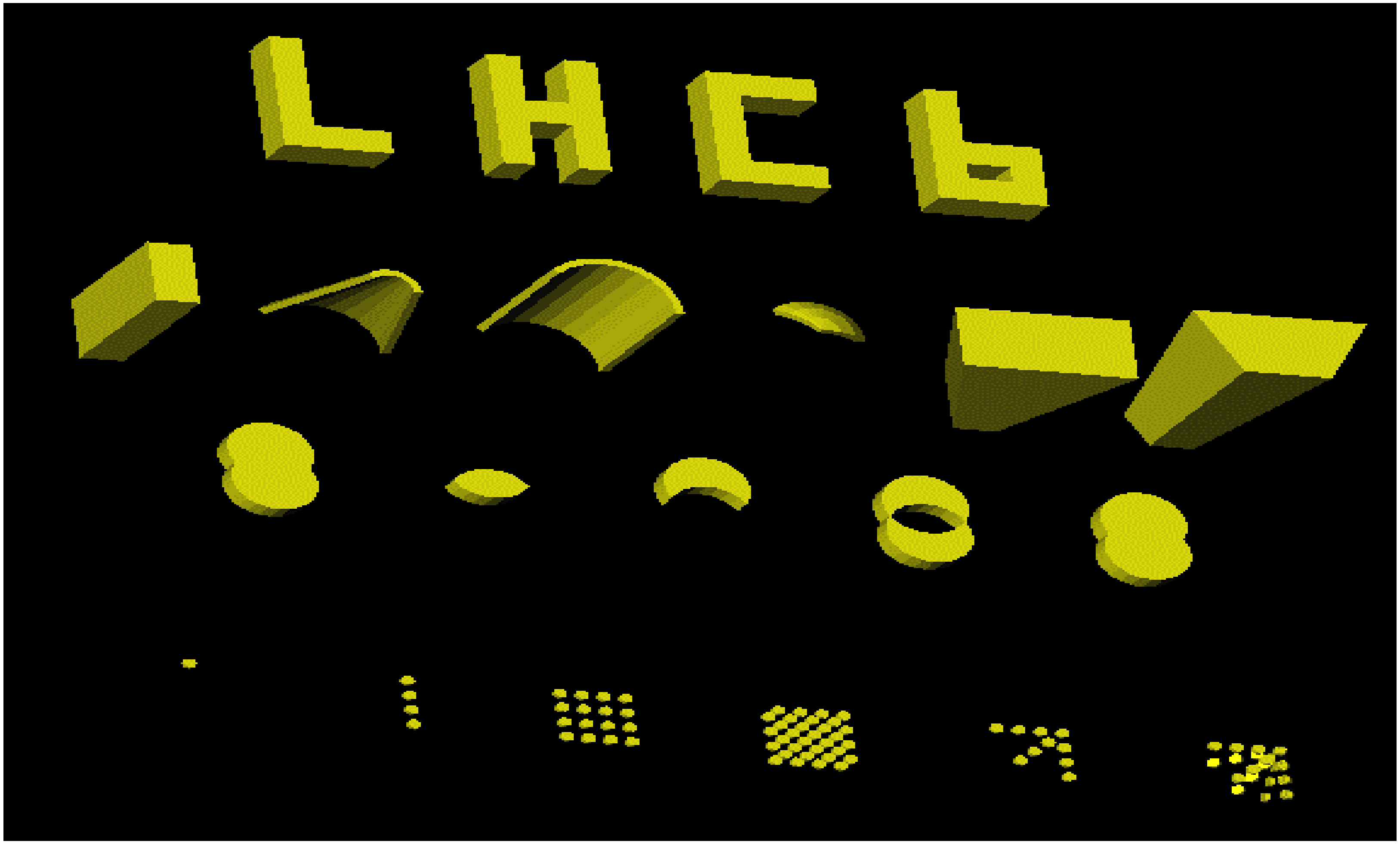
Detector description language news



Topics

- ❑ 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

Default values for attributes

❑ 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

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

Old Way

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

New 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.

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

Old Way

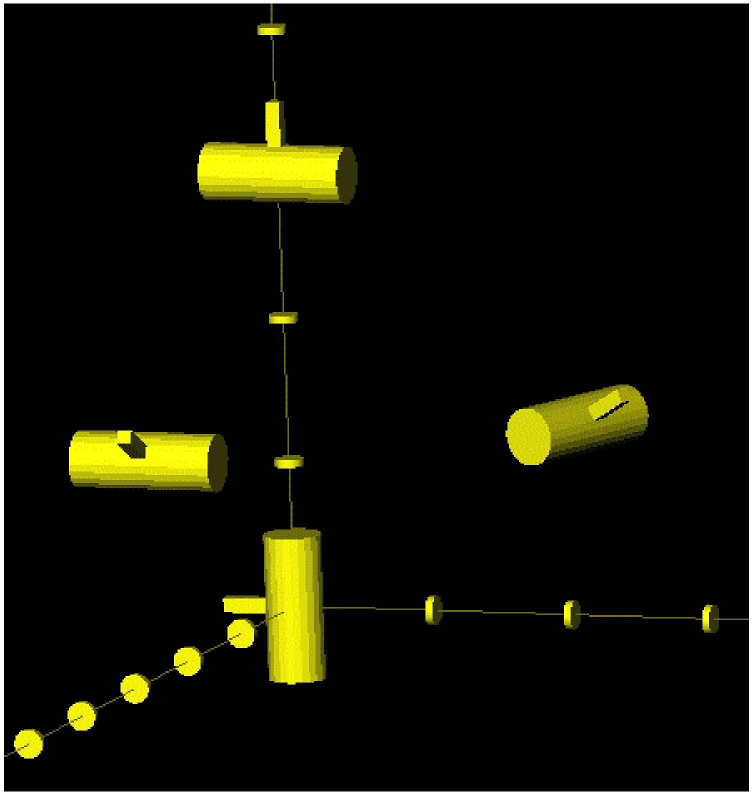
```
<subtraction name="L">
  <box .../>
  <!-- I can put pos and rot here -->
  <box .../>
  <posXYZ .../>
</subtraction>
```

New Way

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

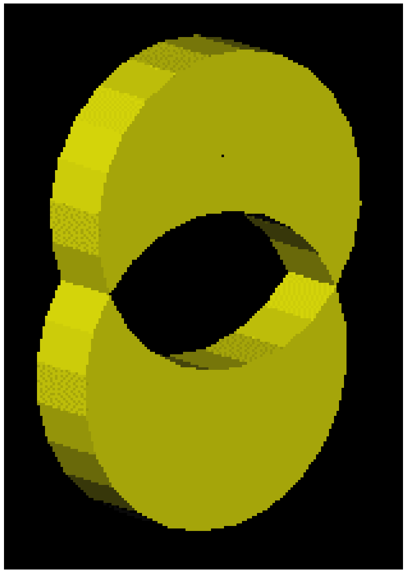
```
<transformation>
  <pos ...>                      |
  <rot ...>                      |   1st
  <transformation>
    <pos ...>                    |   2nd
    <rot ...>                    |
    <pos ...>                    |   3rd
  </transformation>
</transformation>
```



Composition of boolean operations

- ❑ Nothing new but a recursive behavior of booleans.

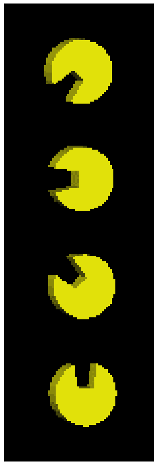
```
<subtraction name="...">
  <union name="...">
    <tubs .../>
    <tubs .../>
    <posXYZ .../>
  </union>
  <intersection name="...">
    <tubs .../>
    <tubs .../>
    <posXYZ .../>
  </intersection>
</subtraction>
```

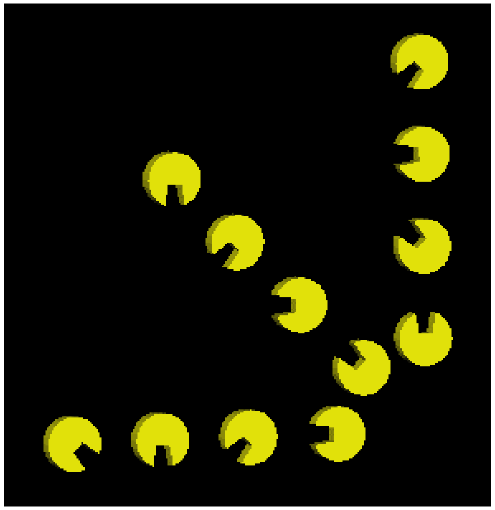


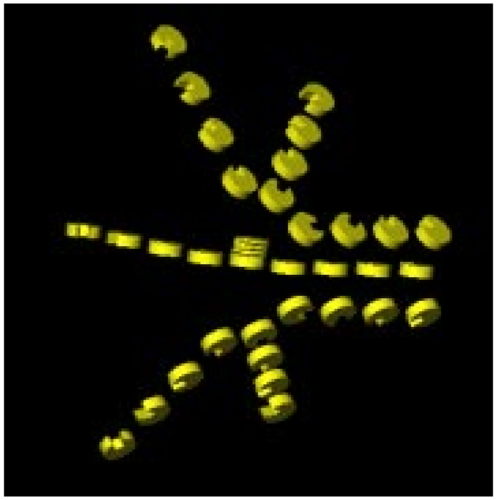
Composition of parametrizations

- Nothing new but a recursive behavior of parametrization.

```
<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 nam... logvol=".../ppv"/>
      <posXYZ/>
      <rotXYZ rotZ="45*degree"/>
   </paramphysvol>
</logvol>
```



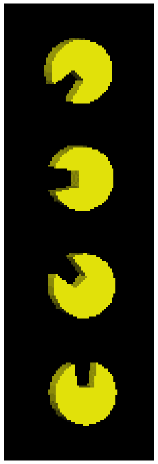


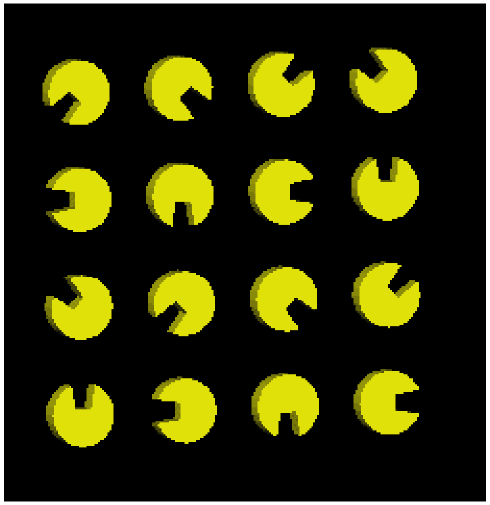


2D and 3D parametrizations

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

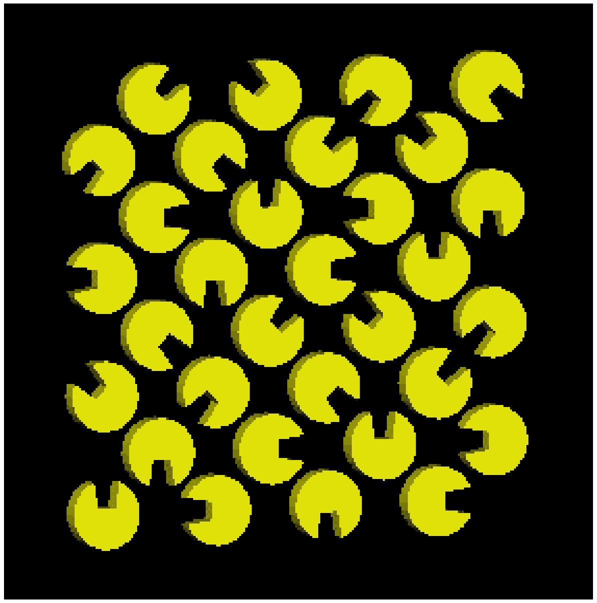
```
<logvol ... name="ppv2D">
  <paramphysvol2D number1="4"
                                    number2="4">
    <physvol ... logvol=".../vol"/>
    <posXYZ y="1*m"/>
    <rotXYZ rotZ="45*degree"/>
    <posXYZ x="1*m"/>
    <rotXYZ rotZ="90*degree"/>
  </paramphysvol2D>
</logvol>
```





Example of 3D parametrization

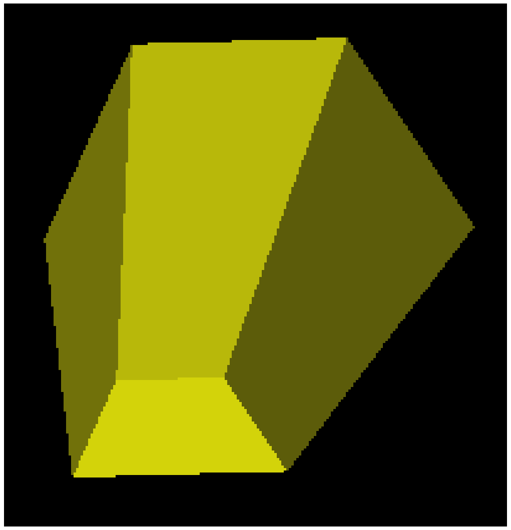
```
<logvol ... name="ppv3D">
  <paramphysvol3D number1="4"
                  number2="4"
                  number3="2">
    <physvol ... logvol=".../vol"/>
    <posXYZ y="1*m"/>
    <rotXYZ rotZ="45*degree"/>
    <posXYZ x="1*m"/>
    <rotXYZ rotZ="90*degree"/>
    <posXYZ x=".5*m" y=".5*m"/>
    <rotXYZ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

```
<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"/>
```



userParameter / userParameterVector

- ❑ 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)

```
<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