Detector description and GeoModel

SPD collaboration meeting

Aytadzh Allakhverdieva

DNLP JINR

05.10.2022

Requirements for SPD geometry description

- Flexibility;
- Ability for description of complicated geometry;
- Geometry version;
- Possibility of using the same geometry description in simulation (Geant4) and reconstruction.

Current version of SPD geometry description

Current version of SPD geometry description is based on ROOT's geometry system TGeo.

Advantage:

- Very flexible.

Disadvantages:

- Non-transparent converting geometry description to Geant4 geometry;
- Each editing of geometry parameters accompanied by compiling all offline software and necessity in interaction with not user-friendly config files;
- Difficult to implement geometry versions.

A Detector Description Toolkits

GeoModel (`https://gitlab.cern.ch/GeoModelDev/GeoModel`)

GeoModel has been used by the ATLAS experiment since 2004.

A toolkit meets all requirements for SPD geometry description.

It doesn't contain magnetic field' description tools.

Documentation is available at `https://geomodel.web.cern.ch/home/`

A Detector Description Toolkits

DD4HEP (`https://gitlab.cern.ch/CLICdp/DetectorSoftware/DD4hep`)

DD4HEP provides a high level of flexibility for the users.

Geometry description also based on using ROOT's geometry system TGeo.

Mainly used for simulation of future experiments.

DD4HEP code looks larger and more complicated with respect to GeoModel.

Geometry and material description in GeoModel

- **an element**: GeoElement* hydrogen = new GeoElement("Hydrogen","H",1.0,1.008*gr/mole);
- **a material**:
  - simple:

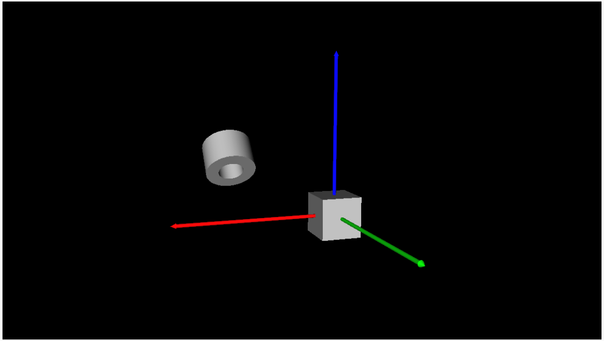
    ```
    GeoMaterial *liquidh = new GeoMaterial("H2", hdensity);
    liquidh→add(hydrogen,1);
    liquidh→lock();
    ```
  - complex:

    ```
    GeoMaterial *ex = new GeoMaterial("Fe+H20", exdensity);
    water→add(water,0.89);
    water→add(iron,0.11);
    water→lock();
    ```

Geometry and material description in GeoModel

- **a shape**:
  GeoBox* box = new GeoBox(length, width, depth);
  GeoModel provides the usage of different types of shapes: boxes, tubes, cones, trapezoids, polycones, etc.
- **a logical volume**:
  GeoLogVol *Logvolume = new GeoLogVol(”LogVol”, box, material);
- **a physical volume**:
  GeoPhysVol* LogPhys = new GeoPhysVol(Logvolume);

Parametrization of physical volumes



```
GeoAlignableTransform *transform = new GeoAlignableTransform(
GeoTrf::RotateX3D(45.0*degree)*GeoTrf::Translate3D(25*cm, 5*cm, 15*cm));
world→ add(transform);
world→ add(tubePhys);
world→ add(cubePhys);
```

The mechanism of interaction with GeoModel

a geometry description

↓

writing the created geometry to a local file (.db, .gdml, .xml)

↓

offline software

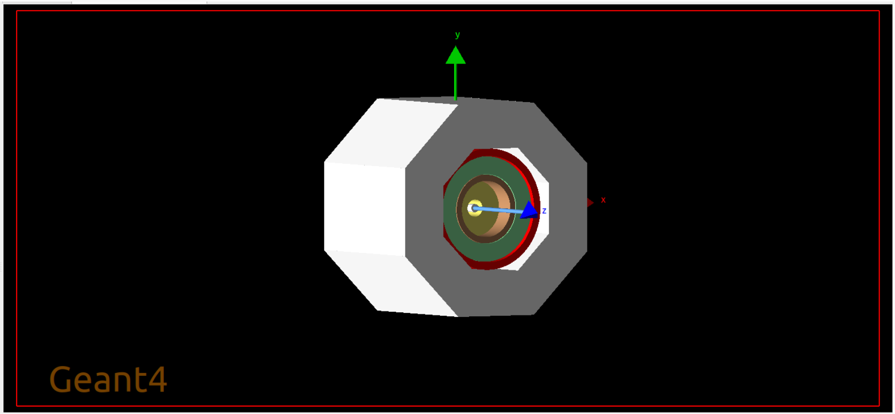
Advantages of the mechanism

- Flexible access to editing the geometry;
- Easier implementation and supporting of geometry versions;
- Optimized disk usage;
- Simple converting geometry description to Geant4 geometry.

Converting GeoModel data to a Geant4 geometry

```
static const std::string path = "/path to db file/spd.db";
GMDBManager* db = new GMDBManager(path);
GeoModelIO::ReadGeoModel readInGeo = GeoModelIO::ReadGeoModel(db);
GeoPhysVol* world = readInGeo.buildGeoModel();
ExtParameterisedVolumeBuilder* builder = new
ExtParameterisedVolumeBuilder("SPD");
G4LogicalVolume* g4World = builder→Build(world);
G4VPhysicalVolume* physWorld = new G4PVPlacement(0, G4ThreeVector(),
g4World, "World", 0, false, 0, true);
```

First step to the full SPD geometry description



Alignment

- Alignment constants is put into the geometry description by using: **setDelta()** method of the GeoAlignableTransform.
- Getting the alignment constants out of GeoModel is through querying a physical volume' transformation.
- Subsystems store alignment information in the calibration database in the form of:
  (Identifier,Transform).

Next steps

- Access and navigation amount geometry objects in reconstruction;
- Association with sensitive volumes of Geant4;
- Straw tracker description.