CAD to Geant4 Conversion

Matti Kalliokoski
BE-BI-BL

MoEDAL software meeting
18/02/15
Geometry Description Markup Language (GDML)

- Application-independent geometry description format based on XML
- Geant4 and ROOT have toolkit bindings for GDML import
  - Not one to one compatible though CAD models can be imported to ROOT and passed to Geant4
- In Geant4 writer is called through G4GDMLParser
  - Requires XercesC parser
G4GDMLParser

- Part of persistency package
  - In Geant4.10 needs to be additionally selected during installation
    - cmake -DGEANT4_USE_GDML=ON
- G4GDMLParser instance dumps the geometry tree to the top volume
- Can be done anywhere in the user code provided only that the Geant4 geometry tree is already instantiated in the memory
- Four examples in examples/extended/persistency/gdml/

Matti Kalliokoski - MoEDAL Software Meeting
CAD to GDML Conversion

• Various options exists depending on preferred CAD tool

• Some examples:
  • FreeCAD to GDML project: http://cad-gdml.in2p3.fr/
    • Not finalized, has some limitations on complex geometries
  • CATIA-GDML geometry builder: http://iopscience.iop.org/1742-6596/331/3/032035/
  • FASTRAD http://www.fastrad.net/,
    • STEP to GDML converter
    • STEP files can be created with most of the CAD tools (AutoCAD Mechanical, CATIA, SolidWorks etc.)
Example: Shielding Box
Example: Shielding Box
Example: Shielding Box

- In GDML export, single or multiple files can be created
  - Single file optimal for small models
  - Separate file for each component better for more complex models
  - Better for modifications
Example: Shielding Box

```cpp
#include "GIGMPLParser.hh"

//...00000000000........00000000000........00000000000........00000000000........

SDetectorConstruction::SDetectorConstruction()
: G4UserDetectorConstruction()
{
    //...00000000000........00000000000........00000000000........00000000000........
    fReadFile="/home/kallikko/workdir/geant4/SIDER-G4MPL/SIDERI/G4MPL_Boxl.Boxl.gmml";
}

//...00000000000........00000000000........00000000000........00000000000........
SDetectorConstruction::~SDetectorConstruction()
{
}

//...00000000000........00000000000........00000000000........00000000000........
G4PhysicalVolume* SDetectorConstruction::Construct()
{
    // Get nist material manager
    G4NistManager* nist = G4NistManager::Instance();
    G4RotationMatrix* rot = new G4RotationMatrix(0,0,0);

    // Option to switch on/off checking of volumes overlaps
    G4bool checkOverlaps = true;
    parser.Read(fReadFile);
    physiWorld = parser.GetWorldVolume();
    G4String Tname = physiWorld->GetName();
    G4cout << Tname << G4endl;
    logicWorld = physiWorld->GetLogicalVolume();
}```
Example: Shielding box
Material description

<?xml version="1.0" encoding="UTF-8"?>
<!-- GDML materials definition from the Fastrad file : "NuSTAR-TPC_310:"-->
- <materials>
  - <element Z="1" formula="VACUUM" name="videRef">
    <atom value="1"/>
  </element>
  - <element Z="1" formula="H" name="hydrogen">
    <atom value="1.00794"/>
  </element>
  - <element Z="2" formula="He" name="helium">
    <atom value="4.002602"/>
  </element>
  - <element Z="3" formula="Li" name="lithium">
    <atom value="6.941"/>
  </element>
  - <element Z="5" formula="B" name="boron">
    <atom value="10.811"/>
  </element>
  - <element Z="6" formula="C" name="carbon">
    <atom value="12.011"/>
  </element>
  - <element Z="7" formula="N" name="nitrogen">
    <atom value="14.00674"/>
  </element>
  - <element Z="8" formula="O" name="oxygen">
    <atom value="15.9994"/>
  </element>
  - <element Z="9" formula="F" name="fluorine">
    <atom value="18.9984032"/>
  </element>
  - <element Z="10" formula="Ne" name="neon">
    <atom value="20.1797"/>
  </element>
  - <material formula="(C22H10N2O5)n" name="Kapton">
    <D value="1.40" unit="g/cm3"/>
    <composite ref="carbon" n="0.52"/>
    <composite ref="hydrogen" n="0.10"/>
    <composite ref="nitrogen" n="0.01"/>
    <composite ref="oxygen" n="0.57"/>
  </material>
  - <material formula="SiO2CaOAl2O3B2O3K2OFe2O3Li2O" name="Stesalite">
    <D value="2.1600" unit="g/cm3"/>
    <fraction ref="SiO2" n="0.668"/>
    <fraction ref="CaO" n="0.05"/>
    <fraction ref="Al2O3" n="0.03"/>
    <fraction ref="B2O3" n="0.22"/>
    <fraction ref="Na2O" n="0.01"/>
    <fraction ref="K2O" n="0.015"/>
    <fraction ref="Li2O" n="0.005"/>
    <fraction ref="Fe2O3" n="0.002"/>
  </material>
  - <material formula="ArCO2" name="ArCO2">
    <D value="0.0018" unit="g/cm3"/>
    <fraction ref="argon" n="0.7000"/>
    <fraction ref="CO2" n="0.3000"/>
  </material>
  - <material formula="C5H4O2" name="Mylar">
    <D value="1.39" unit="g/cm3"/>
    <composite ref="carbon" n="5"/>
    <composite ref="hydrogen" n="4"/>
    <composite ref="oxygen" n="2"/>
  </material>
  - <material formula="AlCOOH" name="AlMylar">
    <D value="1.39" unit="g/cm3"/>
    <fraction ref="Mylar" n="0.8333"/>
    <fraction ref="aluminum" n="0.1667"/>
  </material>
</materials>
More Complex Models

- In simple models it is easy to see if elements are overlapping
  - Will create an error in execution of Geant4
- For more complex models care should be taken in design phase
- More complexity will require more memory in the execution
  - Optimization
    - Remove parts (bolts, washers etc.) that don’t affect the results
Example: Multiwire Proportional Chamber

```cpp
G4VPhysicalVolume* ExHIDetectorConstruction::Construct()
{
    parser.Read(ReadFile);
    physWorld = parser.GetWorldVolume();

    G4String name = physWorld->GetName();
    G4cout << name << G4endl;

    G4LogicalVolume * flogical = physWorld->GetLogicalVolume();
    G4int daughters = flogical->GetDaughters();
    G4cout << "Number of daughters " << daughters << G4endl;

    G4VPhysicalVolume * daughter[10];
    G4VSolid * solidDaughter[10];

    for (G4int i=0; i<daughters; i++)
    {
        G4cout << flogical->GetDaughter(i)->GetName() << G4endl;
        daughter[i] = flogical->GetDaughter(i);
        solidDaughter[i] = daughter[i]->GetLogicalVolume() -> GetSolid();
    }

    flogical->removeDaughter(daughter[0]);
    G4Tube* aTube = new G4Tube("Gas",0.0,solidDaughter[0]->GetExtent().GetXmax(),solidDaughter[0]->GetExtent().GetXmin(),0.0,0.0,0.0,360.0,0.0);
    // G4ThreeVector positionTubel = daughter[0]->GetFrameTranslation();
    G4ThreeVector positionTubel = G4ThreeVector(0.0,0.0,0.0);
    G4LogicalVolume * dlogical = new G4LogicalVolume(aTube,daughter[0]->GetLogicalVolume() -> GetMaterial(),"Gas",0.0,0.0);
    G4RotationMatrix rm = G4RotationMatrix(0.0,0.0);
    rm.rotateY(90.0,0.0);  // deg;
    daughter[0]=new G4VPlacement(G4Transform3D(rm,positionTubel), dlogical, "Gas", flogical, false, 0);

    flogical->removeDaughter(daughter[2]);
    G4Wire* aWire = new G4Wire("CuNuclei",solidDaughter[2]->GetExtent().GetXmax(),solidDaughter[2]->GetExtent().GetXmin(),5000.0,0.0);
    daughter[2]->GetLogicalVolume() -> SetSolid(aWire);
    daughter[2]->GetLogicalVolume() -> AddDaughter(daughter[2]);

    // Note: 5080.0 V is the voltage and 50.0 grams is the tension meaning the weight (for the sag)
    // aWire = new G4Wire("muonTubeWire1", outerRadiusOfTheWire, HalfTubeLength, 5080.0, 50.0);
```
Example: Multiwire Proportional Chamber