Generating a GDML File from the NuMI Geometry

Robert Hatcher
Fermilab Computing Division

2013-05-07

Goal: GDML file

- Desire a GDML representation of the NuMI geometry (as used by g4numi + flugg)
- Code that links to Geant4 libraries (that are built with GDML support) need only add:

```
#include "G4GDMLParser.hh"
...
G4GDMLParser gdml_parser;
G4String outgdml = “myfile.gdml”;
G4VPhysicalVolume* pvol = 0;
gdml_parser.Write(outgdml,pvol);
```

- Alas, that was ... more difficult than one might have thought it should have been...
  - flugg doesn’t link to G4 libraries
  - various versions of G4 didn’t have GDML support or were missing xerces_x (XML parser support)
    - Minerva UPS G4 4.9.2p03 - 3 years old

Using G4StandAlone

- Simple G4 “application” initially for investigating electron shower problem
- Adapted for 4.9.4 + 4.9.5
  - more modifications for 4.9.6 (+ lots of warning messages at run time)
- Procedure

```
setup_nova    # get some version of G4 & ROOT
svn checkout https://cdcvs.fnal.gov/subversion/novaart.pkgs.svn/trunk/G4StandAlone   [note 1]
mkdir G4StandAlone-cmakebuild
cp /nova/app/users/corwin/g4numi121020/numisoft/g4numi/src/*.cc G4StandAlone/src/
cp /nova/app/users/corwin/g4numi121020/numisoft/g4numi/include/*hh G4StandAlone/include/
cp /nova/app/users/corwin/g4numi121020/numisoft/g4numi/src/g4numiCint.h  G4StandAlone/include
cp /nova/app/users/corwin/g4numi121020/numisoft/g4numi/src/g4numiCint.cc  G4StandAlone/src/
rm G4StandAlone/*/NA49*    # don't need the NA49 simulation code
```

.... edits necessary for G4.9.6 + different version of ROOT + newer gcc...

```
cd G4StandAlone-cmakebuild
cmake -DGeant4_DIR=${GEANT4_FQ_DIR}/lib64/Geant4-9.6.1  \
    -DCLHEP_DIR=${CLHEP_DIR}/${CLHEP_FQ}/lib/CLHEP-2.1.3.1  ../G4StandAlone
gmake
./g4sa  --numi --batch -o numi.gdml run_e_1.mac
```

[1] svn+ssh://p-novaart@cdcvs.fnal.gov/cvs/projects/novaart/pkgs.svn/trunk/G4StandAlone

if one might need to commit back changes, one can switch after the fact with

```
svn switch --relocate [old-url] [new-url]
```

GDML output

- Output file has
  - no calculations, just final numbers
  - names have 0xHHHHHHHH address suffix
    - ensures that names are unique ... but a bit annoying
- Fragments

```
<materials>
  <isotope N="1" Z="1" name="H10xb2185b0">
    <atom unit="g/mole" value="1.00782503081372"/>
  </isotope>
  <isotope N="2" Z="1" name="H20xb21c320">
    <atom unit="g/mole" value="2.01410199966617"/>
  </isotope>
  <element name="Hydrogen0xb1f1f40">
    <fraction n="0.999885" ref="H10xb2185b0"/>
    <fraction n="0.000115" ref="H20xb21c320"/>
  </element>
```

GDML output

- Fragments (continued)

```
<solids>
 <box lunit="mm" name="sConcreteSection0and40xb933ea0" x="1508.05" y="854.1" z="52397"/>
 <box lunit="mm" name="sConcreteSection1and30xb933f30" x="933.5" y="4747.9" z="52397"/>
 <box lunit="mm" name="sConcreteSection20xb933fc0" x="6146.9" y="854.1" z="52397"/>
 <box lunit="mm" name="sConcreteSection5Lid0xb934050" x="5842" y="457.2" z="52397"/>
 <box lunit="mm" name="sDuratekBlock0xb934850" x="1330" y="670" z="52397"/>
 <box lunit="mm" name="sDuratekBlockCovering0xb9349a0" x="1355.4" y="665" z="52397"/>
 <tube aunit="deg" deltaphi="360" lunit="mm" name="sBaffle0xb93a360" rmax="30" rmin="5.5" startphi
 <torus aunit="deg" deltaphi="360" lunit="mm" name="sFrontTorus0xb93c8a0" rmax="61.8744" rmin="5
 <box lunit="mm" name="sBox0xb93caa0" x="382.3208" y="382.3208" z="247.4976"/>
 <subtraction name="sHorn1Front0xb93cb20">
  <first ref="sFrontTorus0xb93c8a0"/>
  <second ref="sBox0xb93caa0"/>
  <position name="sHorn1Front0xb93cb20_pos" unit="mm" x="0" y="0" z="123.7488"/>
 </subtraction>
 <polycone aunit="deg" deltaphi="360" lunit="mm" name="sPHorn10C0xb93c730" startphi="0">
  <zplane rmax="157.4546" rmin="149.3266" z="0"/>
  <zplane rmax="157.4546" rmin="149.3266" z="0.013402526270844 8"/>
  <zplane rmax="157.4546" rmin="149.3266" z="19.182895557084 2"/>
  <zplane rmax="206.375" rmin="149.3266" z="19.2028015726926"/>
```

GDML output

- Fragments (continued)

```
 <structure>
 <volume name="lvConcreteSection0and40xb9340e0">
  <materialref ref="Concrete0xb237030"/>
  <solidref ref="sConcreteSection0and40xb933ea0"/>
 </volume>
...
 <volume name="lvPHorn1F0xb94be20">
  <materialref ref="Air0xb220c30"/>
  <solidref ref="sPHorn1F0xb94bc70"/>
  <physvol name="SpiderSupport0xb94ca60">
   <volumeref ref="lvBox0xb94c9a0"/>
   <position name="SpiderSupport0xb94ca60_pos" unit="mm" x="0" y="25.0832669143204" z="489.22
  </physvol>
...

<setup name="Default" version="1.0">
 <world ref="ROCK_log0xb9334a0"/>
</setup>
```

Code Adjustments

- g4numiCint [.h & .cc] are tied to particular ROOT version
  - is this generated, or part of what is distributed?
- G4Exception now takes 4/5 args rather than 1
- gcc 4.7 complains about return type qualifier ‘const’
- diffs: (will attach diff file to DocDB entry)

```
--- /nova/app/users/corwin/g4numi121020/numisoft/g4numi//src/g4numiCint.h    2013-05-06 17:16:58
+++ include/g4numiCint.h      2013-05-06 23:14:14.586517000 -0500
@@ -15,7 +15,8 @@
 #define G__ANSIHEADER
 #define G__DICTIONARY
 #define G__PRIVATE_GVALUE
-#include "cint/G__ci.h"
+//#include "cint/G__ci.h"
+#include "G__ci.h"
 extern "C" {
 extern void G__cpp_setup_tagtableg4numiCint();
 extern void G__cpp_setup_inheritanceg4numiCint();
--- /nova/app/users/corwin/g4numi121020/numisoft/g4numi//include/draytupleMIB_t.hh      2010-08-1
+++ include/draytupleMIB_t.hh      2013-05-06 23:00:34.120721000 -0500
@@ -33,11 +33,11 @@
```