# Geometry Update and To Do List

Ryan Bayes

30 April, 2013

## 1 Status

The CAD compatible implementation of the MICE geometry is in a halfway state. It is possible to run a geometry based on GDML files, but many of those GDML files unwritten or unfinished state. Specifically unwritten geometry files include;

- The TOFs,[1]
- KL,
- Cherenkovs,
- EMR.

The CAD derived geometry does exist but there are some remaining issues including

- a problem in the generation of the geometry producing a "G4Exception : Unknown-InsideOutside" warning — this is an issue in the definition of the Tessellated Solid,
- missing position for the diffuser

The principle problem is the Tessellated Solid problem as the repeated test generated by the "InsideOutside" exception slows loading the geometry into the system dramatically. Tests of simulations with 100 spills and 200 spills (units) indicate that the simulation for 1 spill is 8.135 s with a total geometry loading time of 3 hours 47 minutes.

The current form of the geometry implementation uses the GDML files derived from the CAD files to carry information from the engineering drawings to an implementation of MAUS modules.

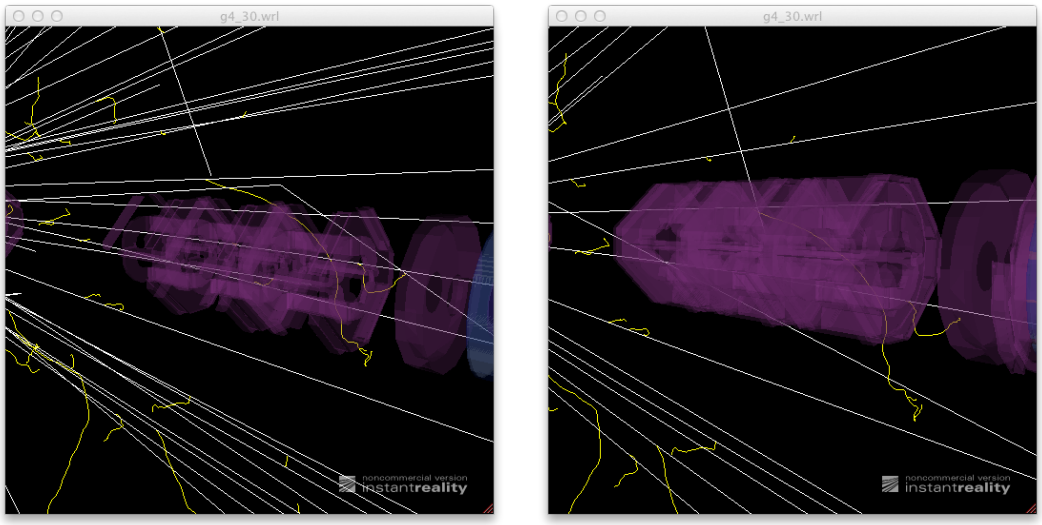
A geometry file for the Tracker has been developed with using the "enhanced" GDML method produced by Matthew. An xslt version mismatch prevents the use of this file in its current form.

[1] The specifications in the legacy files are confirmed correct but the files are half written. An extended GDML implementation has been written with style sheets etc. but only for TOF0. The shielding for TOF1 and TOF2 have not been implemented. The files are not tested. An implementation of TOF0 using base GDML has also been written, and has been tested (failed) but there are doubts that it will work with MAUS modules.

# 2 Required Steps

## 2.1 Tessellated Solids

The priority should be to find the elements of the geometry that produce this exception and fix the issue. A list of the effected elements were generated and the results indicated that the majority of geometry elements are not affected (111 elements affected out of 286 elements in total). This information was not helpful as the source files could not be associated with elements in the CAD files. The affected elements "seem" to be those with an external curved (i.e. convex) surface. This may or may not be a defining feature. Eliminating the affected elements significantly decreases the loading time, but the resulting geometry is extremely "spare" as shown in Fig.**??**. The solution is to generate GDML files with useful file names so that specific elements can be identified and to generate GDML files that can be individually inspected using a toy GEANT4 simulation (such as that in the "extended/persistency/gdml/G01" example).




(a) Reduced geometry

(b) Full Geometry

Figure 1: A subset of the geometry visualization focusing on the quadrupoles. The reduced geometry shows only those elements for which there is no tessellated solid exception.

## 2.2 Material Definitions

Another important problem is that all elements in the raw GDML files have a listed material composition of "Aluminum". This appears to be the result of the material information not being transmitted to FastRAD from the CAD implementation. The CAD files themselves

(to my knowledge) do not themselves communicate this information[2]. The obvious solution is to change the material of the geometry elements by hand. This requires knowledge of a the material of a given geometry element, with the same issues as the above point.

### 2.3 Tracker Geometry

The Tracker geometry development has stopped due to a lack of knowledge about the implementation and the methods. The style sheet is supposed to take a single GDML file and write multiple MAUS modules files based on that GDML file. To do this without updating the "xslt" library (which has some trepidation) requires a rewrite of a file for which I have very little understanding, and delegation is difficult because a lack of knowledge of how to deal with the problem. Chris Heidt has tried to reproduce the problem but communication failed due to lost e-mail. This issue must be resumed.

## 3 Further Steps (Speculative)

### 3.1 Removal of MAUS Modules — Move to Pure GDML

Not sure if tessellated solid problem is entirely an issue of the MAUS modules or the formation of the GDML. To test this a toy simulation should be prepared that simulates each element of the geometry using the GDML and the MAUS modules separately. If the GDML succeeds and the MAUS modules fail, it may be a good idea to move to a geometry implementation that uses GDML directly for simulation. This change requires

- GEANT 4.9.5 or greater (I think)
- The creation of a file directly analogous to "src/simulation/legacy/MICEDetectorConstruction.cc" that invokes GDML files in place of the MAUS modules.

A rewrite of the existing detector geometry (Tracker) will be required. Otherwise there are no other changes that are not otherwise required already.

[2] I looked at example STEP and IGES files and I could not find evidence of material definition within the example. This assumes that such a definition is human readable and does not use some arcane coding. Most of the files do seem human readable.