Contribution ID: **397** Type: **Poster**

mesh2gdml: from CAD to Geant4

Thursday 24 May 2012 13:30 (4h 45m)

The ability to directly import CAD geometries into Geant4 is an often requested feature, despite the recognized limitations of the difficulty in accessing proprietary formats, the mismatch between level of detail in producing a part and simulating it, the often disparate approaches to parent-child relationships and the difficulty in maintaining or assigning material definitions to parts.

Geant4 provides a very rich library of basic geometrical shapes, often referred to as "primitives", plus the ability to define compound geometries via boolean operations. It is therefore capable of supporting extremely complex physical geometries composed of simple primitives. Most CAD systems also incorporate primitive volumes, but their definitions differ between programs and often do not map onto the Geant4 primitives, making the conversion difficult at best. However, one can also define a solid in Geant4 as a volume composed of surface facets. This G4TessellatedSolid can be composed of either triangular or quadrangular facets and therefore provides a mechanism for the programmatic importation of shapes and volumes defined in many CAD systems. In addition to CAD programs, there are very many 3D modeling programs which provide the user with convenient graphical user interfaces to create solid models. Usually aimed at gaming or rendering engines, these could be useful as a front end for a graphical geometry editor. Many output formats are supported, including tesselations. Furthermore, this approach provides a useful solution in cases where the objects are intrinsically irregular, such as biological phantoms.

The main impediment to the importation of CAD files into Geant4 has been their proprietary formats. Some existing solutions target recognized interchange formats such as STEP, but even these formats provide challenges, such as complicated file formats, possible loss of hierarchy or material association and little or no mapping to primitives. Thanks to the proliferation of rapid prototyping and additive manufacturing processes, the surface tesselation language (STL) format is the industrial standard for handling triangulated meshes and is ubiquitous as an export format for both CAD and other 3D modelling software. The format consists of a plain list of three dimensional corner point coordinates (vertex) and flat triangles (facet) with an associated normal vector, making it an ideal candidate for importation into Geant4.

In this talk, we present mesh2gdml, a solution which converts an STL file into a GDML file which can be imported directly into Geant4. The STL facets are translated directly into G4TriangularFacets which are used to create G4TessellatedSolids. Since there is no other structure in an STL file, one has to also solve the problem of creating "topology from a bucket of facets", which we have done. The one area requiring manual intervention is the assignment of material to the newly created solid or solids. Finally, one can either create a world volume from the bounding box of the volume(s) found in the STL file to use standalone within Geant4, or leave the resulting gdml file as individual volumes to aggregate or incorporate into a common world volume later.

In order to benchmark the performance of Geant4 using tesselated volumes, we have written code which allows the exportation of Geant4 geometries in STL format. This geometry is then reimported into Geant4 after being processed with mesh2gdml, allowing us to directly compare the CPU time difference between a geometry composed of primitives and tesselated solids. A side effect of this STL export is the ability to create a real 3D model of the Geant4 geometry on 3D printers. This enables rapid prototyping of parts, direct comparison of the modeled geometry to CAD geometry, communication with colleagues and outreach to the public.

Despite the inherent performance issues related to navigating through geometries composed of many individual facets and the requirement that material be assigned manually to volumes during the translation process, we believe the approach outlined in this talk provides access to a wider range of geometry inputs and will prove to be useful to a number of user communities.

Author: GRAF, Norman Anthony (SLAC National Accelerator Laboratory (US))

Presenter: GRAF, Norman Anthony (SLAC National Accelerator Laboratory (US))

Session Classification: Poster Session

Track Classification: Event Processing (track 2)