

Development and application of CATIA-GDML geometry builder

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 J. Phys.: Conf. Ser. 513 022003

(<http://iopscience.iop.org/1742-6596/513/2/022003>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 194.85.71.37

This content was downloaded on 30/03/2016 at 09:27

Please note that [terms and conditions apply](#).

Development and application of CATIA-GDML geometry builder

S Belogurov^{1,2}, Yu Berchun², A Chernogorov¹, P Malzacher³, E Ovcharenko¹,
V Schetinin^{1,2}

¹ Institute for Theoretical and Experimental Physics, B. Cheremushkinskaya 25,
Moscow, 117218, Russia

² Bauman Moscow State Technical University, 2-nd Baumanskaya str. 5, Moscow,
105005, Russia

³ GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstrasse. 1,
Darmstadt, 64291, Germany

E-mail: belogurov@itep.ru

Abstract. Due to conceptual difference between geometry descriptions in Computer-Aided Design (CAD) systems and particle transport Monte Carlo (MC) codes direct conversion of detector geometry in either direction is not feasible. The paper presents an update on functionality and application practice of the CATIA-GDML geometry builder first introduced at CHEP2010. This set of CATIAv5 tools has been developed for building a MC optimized GEANT4/ROOT compatible geometry based on the existing CAD model. The model can be exported via Geometry Description Markup Language (GDML). The builder allows also import and visualization of GEANT4/ROOT geometries in CATIA. The structure of a GDML file, including replicated volumes, volume assemblies and variables, is mapped into a part specification tree. A dedicated file template, a wide range of primitives, tools for measurement and implicit calculation of parameters, different types of multiple volume instantiation, mirroring, positioning and quality check have been implemented. Several use cases are discussed.

1. Introduction

Design optimization of complex, densely packed systems in severe radiation conditions requires iterative exchange of geometry between the CAD system and the particle transport MC. Geometry representations in CAD systems and GEANT4/ROOT [1,2] are rather different from each other. The difference is twofold: in the description of solid bodies and in the hierarchy of assemblies. For this reason the automated geometry transfer cannot be widely used. In some cases it is possible, but the result is not optimized for simulations and therefore computations are too slow for big assemblies and complex shapes. We are presenting an update on the development and application of the CATIA-GDML geometry builder – a set of tools for manual building of an optimized simulation geometry with necessary level of detail based on an existing CAD model. Earlier version of the Builder with more extensive discussion of the geometry representation issues was presented at CHEP2010 [3]. The geometry is exchanged via GDML [4]. The CAD system CATIA v.5 is widely used in big physical laboratories like CERN, GSI, LNGS, therefore we base our development on it.



2. An approach to MC geometry building from inside CATIA

The goal of the CATIA-GDML geometry builder is to facilitate manual geometry transfer from CATIAv5 to G4/ROOT. We employ powerful measurement, analysis, and design features of various CATIAv5 workbenches to create a MC-compatible representation of an existing engineering assembly. Manual geometry building allows having enough flexibility for creation of the geometry optimal for fast and efficient simulation. There are three principal ideas behind the builder. (1) For the most of G4/ROOT primitives we built parametrized User Defined Features (UDF) in CATIA. (2) We mapped into specification tree of the CATPart file the structure of GDML geometry description i.e. concepts of Boolean solids, logical, physical, mother volumes, volume assemblies, variables and materials. (3) We use a number of template files and VBA macros for facilitation of the MC compatible geometry building in CATIA

The Builder is targeted on physicists who understand what geometry representation and level of detail are optimal for given experimental setup and simulation tasks. For usage of the tool they should get familiar with only limited functionality of CATIA and our plugins.

By default GEANT4 and ROOT read from GDML files the entire geometry and the definition of materials. In our practice it is more convenient to define materials inside the MC package and to be able to select for each subdetector whether it should be read from GDML or from the ROOT file. Such functionality is introduced into cbmroot [5].

It is important to mention that CATIA is flexible enough for mapping any GDML feature into the specification tree and for automation of laborious pieces of the job. The new developments presented here are driven by use cases mostly from the CBM experiment at FAIR.

3. Mapping the G4/ROOT like geometry to CATIA product tree

3.1. Implementation of primitives and Boolean operations

Solid, liquid and gaseous objects are described in particle transport MC as a combination of a shape and a material. Typically shapes are built as Boolean combinations of simple shapes, so called primitives. Description of a shape as a tessellated closed surface is also possible, but it remains out of scope of our Builder. The corresponding concept in GEANT4 is a *Solid*, in ROOT – *TGeoShape*.

We have implemented the following primitives (GEANT4 notations are used) as CATIA UDFs, with the parameter lists equivalent to those in GDML: *G4Box*; *G4Cons*; *G4Ellipsoid*; *G4EllipticalCone*; *G4EllipticalTube*; *G4Orb*; *G4Para*; *G4Sphere*; *G4Torus*; *G4Trap*; *G4Trd*; *G4Tubs*; *G4TwistedBox*; *G4TwistedTrap*; *G4TwistedTrd*; *G4TwistedTubs*. Pilot versions of *G4Polycone* and *G4Polyhedra* are also available. The UDFs are placed into a CATIA catalog named *G4Catalog*.

The Boolean combinations are realized using such CATIA operations as *Add*, *Remove* and *Intersect*. An example of a Boolean solid with the corresponding fragment of the specification tree is shown in the figure 1.

3.2. Implementation of logical volumes

For each logical volume (*LogicalVolume* in GEANT4, *TGeoVolume* in ROOT) one has to create a CATPart file from the G4part template. The part specification tree contains the name of a material as a parameter, *Partbody*, which is a CSG solid standing for the shape, and a publication of the *Partbody* which keeps neither geometry parametrization nor material. It serves for insertion into bigger (mother) *logical volume*.

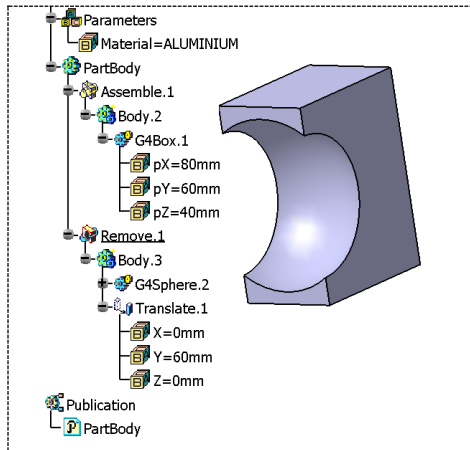


Figure 1. An example of a MC compatible Boolean solid.

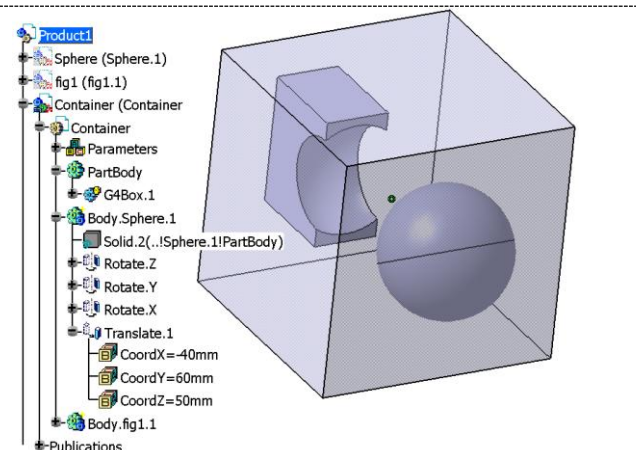


Figure 2. A CATIA representation of a G4Box with two daughter physical volumes inside.

3.2. Implementation of physical volumes, volume assemblies and multiple instantiation

Single and multiple positioning of smaller logical volumes in their mothers or in volume assemblies are performed in the context of a single CATProduct file, where all the CATPart files corresponding to all the logical volumes and volume assemblies are loaded. Publications of smaller volumes from other files can be inserted into CATPart file, corresponding to the mother volume as additional bodies with positioning. Such positioned unparameterized published bodies represent physical volumes (*PhysicalVolume* in GEANT4, *TGeoNode* in ROOT), see e.g. the tree in the figure 2.

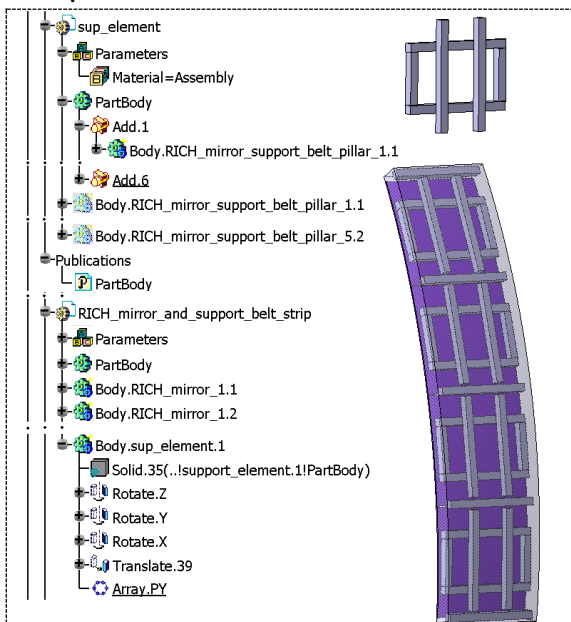


Figure 3. An example of volume assembly (sup_element) and an array of assemblies in the context of CBM RICH mirror segment

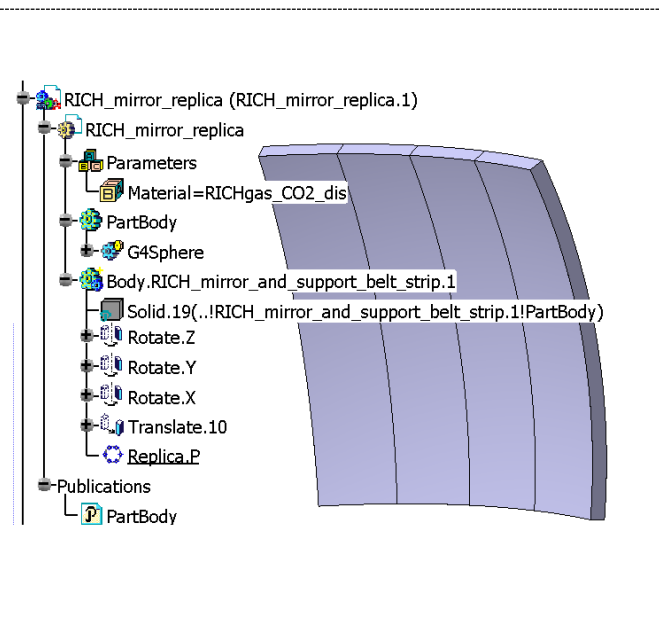


Figure 4. An example of replicated mirror segment in the model of CBM RICH mirror

A special structure of the specification tree is foreseen for the files corresponding to the volume assemblies (*G4AssemblyVolume* in GEANT4, *TGeoVolumeAssembly* in ROOT). The difference between the logical volume and the volume assembly is that the last has neither shape nor material of

the mother – only the reference frame for positioning of the daughters. In this case the value “assembly” is assigned to the string parameter “material“, and the Partbody contains only copies of daughters. It allows the correct visualization in CATIA of multiple level volume assemblies. An example of the volume assembly can be seen in the upper part of the figure 3.

For multiple instantiation of logical volumes in the mother volume we use CATIA Pattern feature in the specification tree. The pattern name can start with either Replica or with Array. In the first case it is mapped to *G4PVReplica* in GEANT4 or to *TGeoVolume * Divide* in ROOT. In the last case the pattern just makes the tree more compact and each instance of the array component is exported into GDML as an individual physical volume. Replicas and arrays can be either circular or linear. Circular arrays are available around X,Y and Z axes, circular replicas only around Z axis. The replicated volumes can be one of the following types: *G4Box*, *G4Sphere* or *G4Tubs*. Replicated slices must fill the mother volume with no gaps. For examples see lower part of the figure 3 and the figure 4.

3.3. Implementation of GDML variables

Geometry exported into GDML can be modified in a convenient way if free parameters are listed as *variables* in the beginning of the GDML file. Any parameters throughout the GDML file can be parameterized with variables and mathematical expressions. CATIA *Parameters* and *Formulas* are used for mapping *variables* into the specification tree.

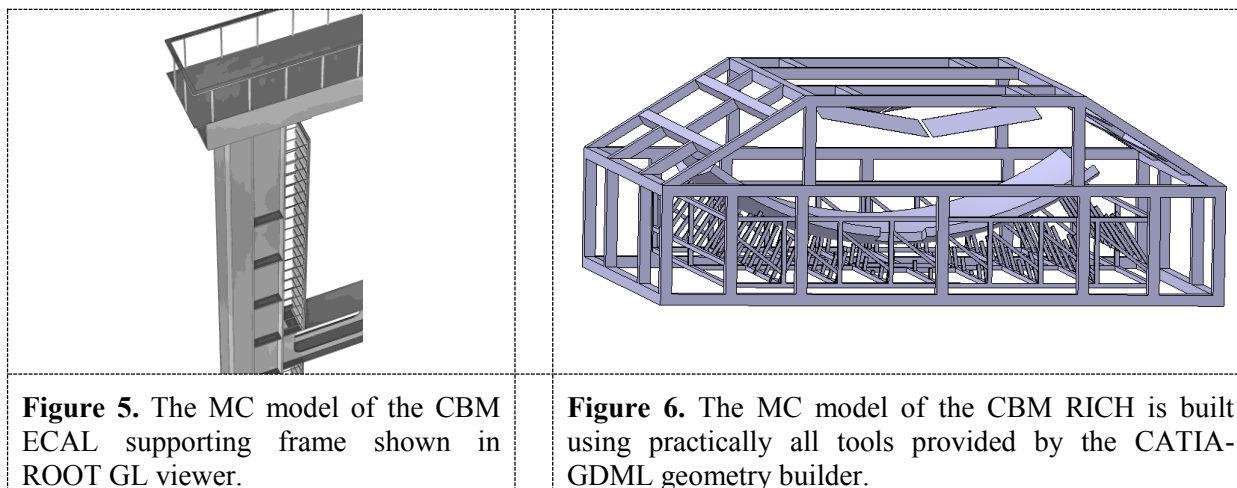
4. Developing the VBA tools facilitating creation of the G4/ROOT like geometry

The original set of macros enabling positioning and replication of the G4/ROOT-like CAD volumes in the corresponding mother volumes was described in [1]. Since then a number of tools facilitating the building procedure itself were developed. The list of macros with short descriptions is shown below.

- *Insertor*: places a daughter volume inside the mother one and creates a template for rotations and translations. A dedicated checkbox allows the user to transform the mother volume into assembly.
- *Array maker*: creates a circular or linear array of individual placements of daughter or “granddaughter” volumes
- *Replica*: generates *G4PVReplica* or *TGeoVolume * Divide*.
- *PtPa*: translates a Boolean operand or a daughter volume by clicking initial and final points.
- *Measure*: allows the user to measure length (radius) of an object or a distance (angle) between two objects and put the result into any parameter of a G4/ROOT like model.
- *Mover*: allows the user to translate simultaneously a set of volumes selected by the mouse.
- *Symmetry*: creates new physical volumes symmetric to the selected ones w.r.t. a given plane. When necessary, a chiral partner of an existing volume is generated. Works so far only with primitives.
- *PickPoint Trap*: calculates and puts into the tree the values of parameters, translations and rotations for a Trapezoid defined by picking with a mouse 8 vertices.
- *Checker*: allows the user to check the correctness of a G4/ROOT compatible tree and overlaps of volumes
- *Material reassigner*: allows the user to assign all the materials from the upper level of the CATIA product.
- *CATIA2GDML*: converts the G4/ROOT compatible tree from CATIA to the GDML file readable by ROOT and GEANT4.
- *GDML2CATIA*: converts geometry from a GDML file to CATIA. It allows bringing any simulation geometry into CATIA. Currently only two hierarchical levels of volumes are visible in each file. However an extension of the approach used for implementation of volume assemblies will allow making this tool fully automated.

5. Examples

Two examples of the CATIA-GDML geometry builder application are shown in the figures 5 and 6.



6. Plans for further development

Further developments of the Geometry Builder are foreseen in the following directions:

- Finalizing the implementation of the primitives *G4Polycone* and *G4Polyhedra*
- Extension of the pickpoint definition to other primitives and handling of inaccuracies
- VBA macros code refactoring and optimization
- Visualization of several hierarchical levels in CATIA
- Automatic fit of parametrized CSG models to existing parts
- Case study and best practice elaboration

7. Conclusions

A CATIA-GDML geometry builder is developed and used for CBM and other FAIR experiments. The Builder is a set of tools for building simulation-optimized geometry for GEANT4/ROOT from inside the CATIAv5 CAD system. The geometry building process is facilitated significantly with respect to the traditional approach by a number of VBA macros. The geometry can be exchanged with detector simulation MC packages via GDML files in both directions. Further development of the builder is possible following requirements of complicated use cases.

Acknowledgements

The authors are grateful I. Ovsyannikov (BMSTU) for his contribution into the project. This work is supported by SC ROSATOM and Helmholtz Association (grant IK-RU-002) via FAIR-Russia Research Center.

References

- [1] Allison J et al. 2006 Geant4 developments and applications *IEEE Trans. on Nucl. Sci.* **53** 270-8
- [2] Naumann A, Offermann E, Onuchin V, Panacek S, Rademakers F, Russo P and Tadel M 2009 ROOT – a C++ framework for petabyte data storage, statistical analysis and visualization *Computer Physics Communications* **180** 2499-512
- [3] S. Belogurov, Yu. Berchun, A.Chernogorov, P. Malzacher, E. Ovcharenko ‘CATIA-GDML geometry builder’, *J. Phys.: Conf. Series*, V. 331 (2011) 32035
- [4] Chytrcek R, McCormick J, Pokorski W and Santin G 2006 Geometry description markup language for physics simulation and analysis applications *IEEE Trans. Nucl. Sci.* **53** 2892-6
- [5] F. Uhlig, M. Al-Turany, D. Bertini, R. Karabowicz Software Development Infrastructure for the FAIR Experiments *J. Phys: Conf. Series* V. **331** (2011) 042024