

The Virtual Geometry Model

Ivana Hřivnáčová^{1,*}

¹Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France

Abstract. The Virtual Geometry Model (VGM) is a geometry conversion tool, providing conversion between Geant4 and ROOT TGeo geometry models. Its design allows inclusion of another geometry model by implementing a single sub-module instead of writing bilateral converters for all already supported models.

In this presentation we will give an update on the tool architecture, implementation and supported features, the user examples, testing and documentation. We will discuss the opportunities for using the tool for verification of the user geometries. Finally, we will also present the tool build system, distribution and releases policy.

1 Introduction

The Virtual Geometry Model (VGM) is a geometry conversion tool that provides the conversion between Geant4 [1] and ROOT [2] geometry models and the export of geometry into two XML formats: GDML [3] and AGDD [4] (see Figure 1). Its design allows inclusion of another geometry model by implementing a single sub-module instead of writing bilateral converters for all already supported models.

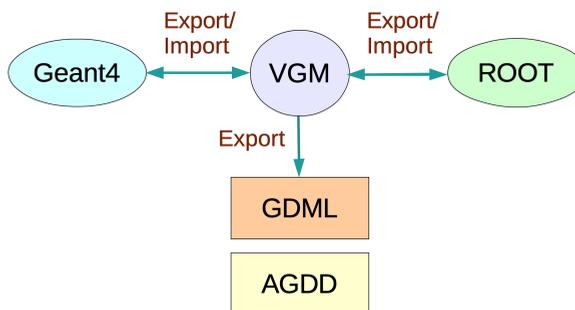


Figure 1. VGM functions overview.

The VGM was last presented at CHEP in 2007 [5] and since then it has been under a continuous maintenance and development, following the evolutions of the supported geometry models and adapting to different use cases. It was implemented in the context of Geant4

*e-mail: ivana@ipno.in2p3.fr

Virtual Monte Carlo (Geant4 VMC) [6], where it is used for the support of the TGeo [7] geometry definition with the Geant4 native geometry navigation or the support of the Geant4 geometry definition with the Geant3 TGeo navigation. At present, it is used in large experimental frameworks, such as FairRoot [8] and ALICE O2 [9]. In FAIR, the VGM is used by the PANDA experiment [10] in the context of Geant4 VMC for the support of the TGeo geometry definition with the Geant4 native geometry navigation, and in ALICE O2 for geometry verifications.

2 VGM Components

The core of VGM consists of a set of abstract interfaces provided in pure header files with no external dependencies. The interfaces are implemented for two supported geometry models with use of common implementation provided in two utility libraries. Besides the geometry model specific packages, VGM also provides the XML exporter, that allows to export the geometry of the supported modeller in two XML formats: GDML and AGDD.

The design presented in the previous papers was stable in time. The class diagrams presented in [5] remain valid with only one exception. After the suppression of the AgddGM implementation, the development of which was stopped in 2010, only two concrete implementations of the interfaces, Geant4GM and RootGM, are available.

Below we give an overview of the VGM components presented in Figure 2:

- Interfaces (VGM) - the abstract interfaces to geometry entities used for geometry description (solid, volume, placement, material) and to the factories for creating geometries and their import/export
- Utilities (BaseVGM, ClhepVGM) - the common implementation and utility functions
- Geometry modules (Geant4GM, RootGM) - the implementation of the VGM interfaces for concrete geometry models
- XML Exporter (Xm1VGM) - the exporters to GDML and AGDD XML formats
- Examples - four examples demonstrating a geometry conversion (E01, E02) and an export into XML (E03, E04)
- Test - the test application covering most of supported use cases

3 Supported features

In general, VGM supports the intersection of the features of ROOT and Geant4 geometries, plus some features existing in one geometry model and implemented with work-arounds in the other geometry model.

The overview of common supported solids is presented in Figure 3 with shortened self-descriptive names. These solids can be also used with applied transformations in the form of scaled, displaced or Boolean solids. With the exception of Geant4 parameterised volumes, which cannot be supported in a general way, VGM supports all types of volume placements: from simple placements with a transformation including translation, rotation and reflection to multiple placements such as divisions, replicas and assemblies. Recently, the support for Geant4 replicated slices (divisions with gaps) was implemented with a work-around implementation for the ROOT geometry model.

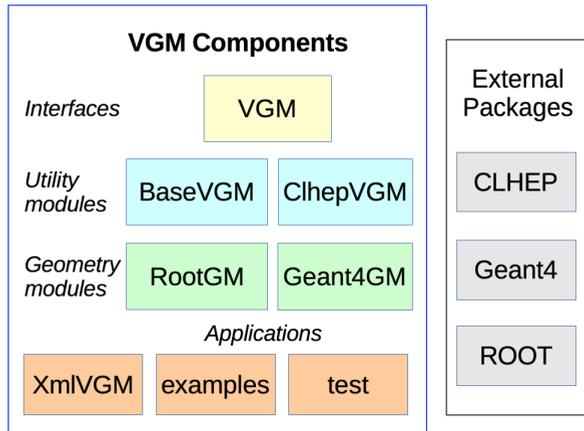


Figure 2. VGM Components Overview

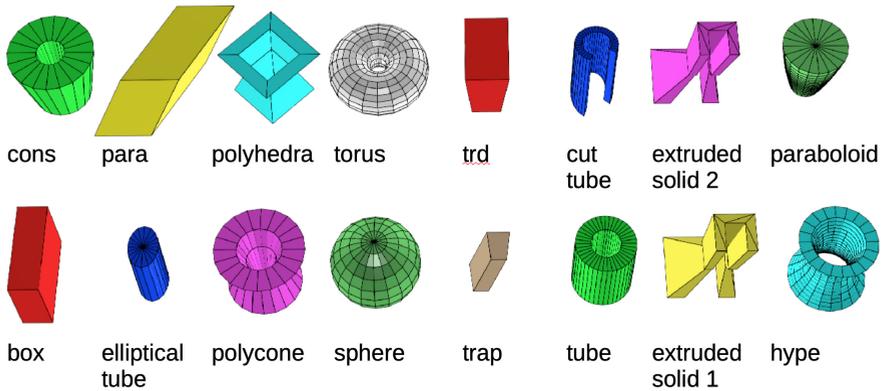


Figure 3. Overview of solids supported in VGM

4 Use cases

A fast, in-memory bi-lateral conversion between the supported geometry models, which motivated this tool development in the past, remains the main use case of the tool today. Meanwhile several other ones also have found their place.

The integrated test program, with a large variety of run options, illustrated in Listing 1, provides an excellent tool for the verification of complex geometries. The 'Special' test option can be used to import an external ROOT geometry, export it in Geant4 and then shoot a number of geantinos in random positions and directions, transport them with Geant4 native or G4Root navigation in source and target geometries and finally compare the outputs from both tests.

```
Usage:
vgm_test inputType inputFactory outputFactory outputXML selectedTest
[debug] [openAngle] [noVis] {run} [rootNavig]
inputType      = VGM, AGDD, Geant4, Root
inputFactory   = AGDD, Geant4, Root
```

```

outputFactory = Geant4, Root, None
outputXML     = AGDD, GDML, noXML
selectedTest  = Solids, NewSolid, Placements[2], Reflections, Assemblies,
               AssembliesN, BooleanSolidsM, ScaledSolids,
               Special, DisplacedSolidN
               where N = 1, 2; M = 1, 2, 3
debug         = if specified the factories operate in debug mode
ignore       = if specified the factories operate in ignore mode
bestMatch    = if specified the factories operate in bestMatch mode
openAngle    = if specified, solids like tubs, cons etc. are built
               with open azimuthal angle
noVis        = no visualisation
run          = run with the generator defined in macro/selectedTest.mac
               rootNavig = use navigation via G4Root
               singleMode = activate single mode conversion
                   (only convert solids from built geometry one by one)

```

Listing 1. The vgm_test program options (printed by the program when called with wrong arguments)

By default, when the source geometry contains an incompatible feature, for example an "exotic" shape type present only in the source geometry model, the program stops, printing detailed information about the encountered incompatibility. Users can set the 'ignore' option to prevent the program from stopping. Then, all unsupported solids will be replaced with boxes of small dimensions, which can be further customized. In this way all features blocking a geometry conversion can be evaluated at once. This option has also been found useful by the users for exporting their Geant4 geometry in ROOT and using the ROOT visualization despite the geometries incompatibilities. The 'debug' option can be used to dump the parameters of all geometry objects in a user-friendly readable format.

The export into XML, useful to interchange geometries with other tools, turned out to be less important when the native GDML import and export became reliable with both geometry models. It can however be of interest in case of extending VGM with another geometry model without the GDML support, or also as an alternative in case of a failure of the export with the native tool. The AGDD format remains interesting for the explicitly written transformations being part of the volumes compositions, and so easy to find when investigating geometry problems like volumes overlaps or protrusions.

A single conversion mode was implemented according to a requirement from the VecGeom project. While the VGM was developed for dealing with the whole volume hierarchies starting from the top (world) volume, this mode allows to activate a conversion of a single object. At present, this mode can be used only with solids, but it can be extended for other object types (e.g. materials, volumes) if needed.

4.1 Examples

Besides the complex test program, being written in particular for the purpose of testing all aspects, four simple examples demonstrating the use of VGM are provided. Two examples (E01, E02) demonstrate a geometry conversion between Geant4 and Root geometries and the remaining two examples (E03, E04) an export from Geant4 and Root geometries into XML.

4.2 Test suites

VGM testing is based on shell scripts which automatically run 20 different geometries, each one in 20 different test configurations. In addition to the six geometries already described in [5], new test cases were gradually added to cover new features (scaled and displaced solids, assemblies and boolean solids with positions with reflections, placements via a replica slice)

or use cases where a failure in either one of the geometry models or in the conversion itself was fixed. The reference test outputs are provided with the source code and updated with each release.

5 Development, build system and deployment tools

In 2012, the VGM build system moved to CMake and, since version 4.0 in 2014, the support for the old build systems (GNUmake based, Autoconf and CMT) was stopped. The Autoconf and CMT configuration files were removed and the GNUmake based files were kept with no further maintenance and will be removed in future.

As VGM was implemented in the context of Geant4 VMC, in 2017, it was moved from SourceForge into GitHub [11] to become part of the VMC project packages in the `vmc-project` organization [12].

The usage of clang-format tool [13] has been introduced recently to automatically format the code in order to obtain a consistent code style with the other `vmc-project` packages. A configuration file with a customized style and a script for automatically applying the tool were introduced.

With each new version, the source code documentation is automatically generated with the Doxygen [14] tool and the release notes are presented with a detailed description of the new developments and bug fixes in the dedicated "history" file on the Web site [15].

6 Conclusion

VGM is a fast, in-memory, geometry conversion tool which has been in production since 2004 and, at present, is part of the external software used in large experimental frameworks, such as FairRoot or ALICE O2. In this paper we gave an update on its evolution in the last decade and presented its components, new features and use cases. We also briefly discussed the tools used for the development, build system and deployment.

References

- [1] Agostinelli S et al 2003 *Nucl. Instrum. and Methods* **A506** 250-303
Allison J et al 2006 *IEEE Transactions on Nuclear Science* **53 No. 1** 270-278
Allison J et al 2016 *Nuclear Instruments and Methods in Physics Research A* **835** 186-225
- [2] Brun R et al 1997 *Nucl. Inst. & Meth. in Phys. Res. A* **389**, 81-86
<http://root.cern.ch>
- [3] Chytráček R 2001 *Proceedings of the CHEP2001 conference* Id 8-009
- [4] Arnault Ch et al 2001 *Proceedings of the CHEP2001 conference* Id 8-001
- [5] Hřivnáčová I 2008 *J. Phys: Conf. Series* **119** 042016
- [6] https://github.com/vmc-project/geant4_vmc
- [7] Brun R, Gheata A et al 2003 *Proceedings of the CHEP2003 conference* pp THMT001
- [8] <https://fairroot.gsi.de>
- [9] <https://alice-o2-project.web.cern.ch>
- [10] <https://panda.gsi.de>
- [11] <https://github.com>
- [12] <https://vmc-project.github.io>
- [13] <https://clang.llvm.org/docs/ClangFormat.html>
- [14] <http://www.doxygen.nl>
- [15] <https://github.com/vmc-project/vgm/blob/master/doc/VGMhistory.txt>