

A General Geometry System

BV

[2014-05-29 Thu 17:21]

1 Overview

Here, "geometry" refers to the description of the material makeup of a region of space for the purpose of simulating or reconstructing the trajectory of particles through the space. This document describes the components of a geometry system, presents a survey of existing (partial) systems and finally outlines one plan for a complete system.

2 Components

required for a system that can produce, manage, and consume representations of the geometry information and includes its validation and visualization and conversion between different representations.

A geometry system can be broken down into the following components:

conceptual schema the meta-description of the data types used to describe any particular geometry

representations a description may be represented in various forms: persistent files, transient, in-memory collection or visually as a 2D or 3D static or interactive rendering.

converters the processes by which one representation may be converted into another.

producers mechanisms and tools to create or alter a description, usually through visual or transient representations and resulting in persistent ones

validation mechanisms and tools which assure some level of correctness of a description, typically by operating on a transient representation or by human inspection through visualization.

management a mechanism by which a description may undergo controlled and recorded change.

All existing geometry systems have at least one of these components (by construction). Few, if any, have all. Inter-separability between existing systems allows some level of aggregation of components into a full system but this still leads to incomplete, inefficient or awkward results.

2.1 Conceptual schema

All existing schema considered here share a large overlap in schema at the conceptual level. They all use a "solid model" as apposed to a "surface model" for their description. This model contains these main data objects:

shape a.k.a. "solid" is a region of space in a specific shape and size with a conventionally chosen coordinate system.

material the collection of bulk physical properties (density, isotopes, optical properties, sometimes visualization properties).

boundary a property related to the boundary of two materials, but not associated with either (for example, to model reflection in a non-Lorentz way)

volume the aggregation of a shape, material and possibly boundaries as well as other volumes. A sub-volume is placed w.r.t. a volume by specifying a translation and rotation. Typically any sub-volume shapes must be completely contained in the volume shape and no sub-volume shapes may overlap. This sub-volume placements produce a volume hierarchy.

touchable a path through the volume hierarchy in order to indicate some semantic importance to a particular region of space.

With these few concepts a description may be made which is adequate for many purposes. Additional important extensions are needed to provide optimization by exploiting symmetries that may exist in the geometry being described. These include:

replication repeated placing of identical sub-volumes. Note, this is not necessarily the same as representation-level looping constructs.

FIXME: what other schema elements are missing?

2.2 Representations

There are three forms of representation of conceptual schema that are important:

2.3 Survey of important, existing geometry systems

There are a few existing geometry systems which are important to High Energy Physics (HEP) and other fields. Some are deeply integrated with important tools and their support must be retained. Others are used in ancillary or uncommon tools but provide useful features and functionality. This section lists these systems or elements of them and calls out some beneficial features or perceived problems in their design or their common use. The list is separated into three major categories of functionality.

2.3.1 Persistent Representations

This section lists common persistent representations (ie, file formats) for storing geometry information which exist in HEP.

1. TGeo ROOT files

The `TGeo` classes part of the ROOT data analysis framework may be saved into a compressed, binary and platform independent format ("ROOT file"). Instances of these classes must be created and aggregated through via specially written software. See more information in the section on 2.3.3 and 2.3.2.

2. GDML XML

The Geometry Description Markup Language (GDML) is a text based format conforming to an XML schema. Besides allowing for exhaustive geometry description it has language structures for defining constant and variable parameters, general loops as well as explicitly replicated volumes. The geometry information may be spread among various GDML XML files and then aggregated through the standard XML external entity feature.

3. AGDD XML

The ATLAS Generic Detector Description (AGDD) is a text based format conforming to an XML schema. Like GDML it allows exhaustive geometry description as well as language structures for defining constant and variable parameters, loops, arrays. Files can be aggregated via the standard XInclude mechanism. Beyond a physical geometry description, AGDD has support for visualization by allowing colors to be defined on volumes.

4. DetDesc XML

The Detector Description (DetDesc) is a text based format conforming to an XML DTD. Like GDML and AGDD it allows exhaustive geometry description as well as language structures for defining constant and variable parameters, loops and vectors. It allows aggregation through the standard XML external entity mechanism as well as a unique reference mechanism analogous to and named identically to HTML's `href`. It has been extended to allow for injection of XML at run-time through environment variables.

2.3.2 Transient Representations

2.3.3 Conversion Tools

This section lists tools that can convert from one form to another be they persistent or transient.

1. VGM
2. ROOT
3. Geant4 Geometry XML Converter
<http://root.cern.ch/root/vmc/XML.html>
4. DetDescCnv
5. GiGaCnv

2.3.4 Visualization

There are many such systems in existence, such as those that are part of Geant4 or ROOT. There are conversion formats such as GDML and AGDD and parsers/converters that handle them such as

as well as experiment-specific systems like DetDesc written by LHCb and used by others such as Daya Bay. There are also application specific

3 Links

Maxim's agdd/gdml comparison http://www.star.bnl.gov/public/comp/simu/GDML/gdml_vs_agdd.html

Maxim's STAR simulation pages <http://www.star.bnl.gov/public/comp/simu/newsite/>