Introduction to Geant4

Version: geant4 9.2

Published 19 December, 2008

Geant4 Collaboration

Introduction to Geant4

by Geant4 Collaboration

Version: geant4 9.2

Published 19 December, 2008

Table of Contents

1. Geant4 Scope of Application 1
2. History of Geant4 2
3. Overview of Geant4 Functionality 3
4. Geant4 User Support 5
5. Software Knowledge Required to Use the Geant4 Toolkit 6
6. Computing Environment Required by the Geant4 Toolkit 7

Chapter 1. Geant4 Scope of Application

Geant4 is a free software package composed of tools which can be used to accurately simulate the passage of particles through matter. All aspects of the simulation process have been included in the toolkit:

- the geometry of the system,
- the materials involved,
- the fundamental particles of interest,
- the generation of primary events,
- the tracking of particles through materials and electromagnetic fields,
- the physics processes governing particle interactions,
- the response of sensitive detector components,
- the generation of event data,
- the storage of events and tracks,
- the visualization of the detector and particle trajectories, and
- the capture and analysis of simulation data at different levels of detail and refinement.

Users may construct stand-alone applications or applications built upon another object-oriented framework. In either case the toolkit will support them from the initial problem definition to the production of results and graphics for publication. To this end, the toolkit includes:

- user interfaces,
- built-in steering routines, and
- command interpreters

which operate at every level of the simulation.

At the heart of Geant4 is an abundant set of physics models to handle the interactions of particles with matter across a very wide energy range. Data and expertise have been drawn from many sources around the world and in this respect, Geant4 acts as a repository which incorporates a large part of all that is known about particle interactions.

Geant4 is written in C++ and exploits advanced software-engineering techniques and object-oriented technology to achieve transparency. For example, the way in which cross sections are input or computed is separated from the way in which they are used or accessed. The user can overload both of these features. Similarly, the computation of the final state can be divided into alternative or complementary models, according to the energy range, the particle type, and the material. To build a specific application the user-physicist chooses from among these options and implements code in user action classes supplied by the toolkit. A serious problem with previous simulation codes was the difficulty of adding new or variant physics models; development was difficult due to the increased size, complexity and interdependency of the procedure-based code. In contrast, object-oriented methods help manage complexity and limit dependencies by defining a uniform interface and common organizational principles for all physics models. Within this framework the functionality of models can be more easily recognized and understood, and the creation and addition of new models is a well-defined procedure that entails little or no modification to the existing code.

Chapter 2. History of Geant4

These ideas first appeared in two studies done independently at CERN and KEK in 1993. Both groups sought to investigate how modern computing techniques could be applied to improve the existing FORTRAN based Geant3 simulation program. Activities were merged in the fall of 1994 and a formal proposal, RD44, to construct an entirely new program based on object-oriented technology was submitted to CERN's Detector Research and Development Committee. The initiative grew to become a large international collaboration of physicist programmers and software engineers from a number of institutes and universities participating in a range of high-energy physics experiments in Europe, Japan, Canada and the United States. The objective was to write a detector simulation program which had the functionality and flexibility necessary to meet the requirements of the next generation of subatomic physics experiments. The initial scope quickly widened when it became apparent that such a tool would also benefit the nuclear, accelerator, space and medical physics community, with more individuals joining from these fields of science.

The RD44 project represented a pioneering effort in redesigning a major CERN software package for a modern object-oriented (OO) environment based on C++. The R&D phase was completed in December 1998 with the delivery of the first production release. The collaboration was subsequently renamed Geant4 and reinstated on the basis of a formal Memorandum of Understanding (MoU) signed by many of the same national institutes, laboratories and large HEP experiments who participated in RD44. The agreement addresses the program management, maintenance and user support during the production phase and the continued development and refinement of the toolkit. It is subject to tacit renewal every two years and sets out a collaboration structure defined by a Collaboration Board (CB), a Technical Steering Board (TSB) and several working groups. In February 2006 a new MoU came into effect and CB and TSB were renamed to Oversight Board (OB) and Steering Board (SB) respectively.

The collaboration now profits from the accumulated experience of many contributors to the field of Monte Carlo simulation of physics detectors and physical processes. While geographically distributed software development and large-scale object-oriented systems are no longer a novelty, Geant4, in terms of the size and scope of the code and the number of contributors, may well represent the largest and most ambitious project of its kind outside the corporate world. A clean overall problem decomposition has led to a clear hierarchical structure of domains. Every section of the Geant4 software, which corresponds to a releasable component (library), is individually managed by a working group of experts. In addition, there is a working group for each of the activities: testing and quality assurance, software management and documentation management. A release coordinator heads each group. This consequent distribution of responsibility among a relative large number of people permits a support structure whereby outside users can address questions directly to the appropriate expert.

Chapter 3. Overview of Geant4 Functionality

The Geant4 class category diagram is shown in Figure 3.1

Categories at the bottom of the diagram are used by virtually all higher categories and provide the foundation of the toolkit.

The

- *global*

category covers the system of units, constants, numerics and random number handling.

The two categories:

- *materials*
- *particles*

implement facilities necessary to describe the physical properties of particles and materials for the simulation of particle-matter interactions.

The

- *geometry*

module offers the ability to describe a geometrical structure and propagate particles efficiently through it.

Figure 3.1. Geant4 class categories

Above these reside categories required to describe the tracking of particles and the physical processes they undergo. The

- *track*

category contains classes for tracks and steps, used by the

- *processes*

category, which contains implementations of models of physical interactions: electromagnetic interactions of leptons, photons, hadrons and ions, and hadronic interactions.

All processes are invoked by the

- *tracking*

category, which manages their contribution to the evolution of a track's state and provides information in sensitive volumes for hits and digitization.

Above these the

- *event*

category manages events in terms of their tracks and the

- *run*

category manages collections of events that share a common beam and detector implementation. A

- *readout*

category allows the handling of pile-up.

Finally capabilities that use all of these categories and connect to facilities outside the toolkit through abstract interfaces, provide *visualization*, *persistency* and user *interface* capabilities.

Chapter 4. Geant4 User Support

The collaboration offers support for Geant4, providing

- assistance with problems relating to the code,
- consultation on using the toolkit, and
- response to enhancement requests.

A user can also expect assistance in

- investigating aberrant results.

Users of the software who encounter a problem in running the code can use an

- Internet-based problem reporting system .

The system is open to all users. It is set up automatically to assign problem reports to the responsible person according to the category affected. The contact person may then respond directly or forward it to a colleague. This system is a customized version of the open source reporting tool Bugzilla . Besides routing the problem to specialists, it tracks and documents the responses until the problem is resolved.

New requirements, such as requests for new functionality, are presented to and decided by the Steering Board (SB). The SB sets the priorities and agrees on time-scales for the fulfillment of new requirements. Such support is guaranteed to collaboration members, while requests from non-members are handled on a *best effort* basis.

For each member organisation a contact person (SB member) has been designated who acts as a first reference for Geant4 users in that locality, which may include affiliated institutions, user groups, and others in the same geographic area. The contact person will respond to enquiries, help resolve simple problems, and forward more specialized queries to the relevant expert(s).

Beyond that, a list of frequently asked questions (FAQs) , and an internet-based user forum complete the available Geant4 user support.

Chapter 5. Software Knowledge Required to Use the Geant4 Toolkit

In general, there are three types of users:

- the **end user**,
- the **application programmer**,

and for large simulation tasks:

- the **framework provider**.

The **end user** runs the simulation program by controlling run time parameters. The interface with the program may be a graphical user interface, an interactive command line interface, or the macro-based system for batch. The end user needs a basic knowledge of how to control the program flow but does not necessarily have to know object-oriented programming or C++.

The **application programmer** is central to any simulation task. A firm knowledge of C++ is required to implement code in user action classes to specify, at a minimum, the detector description, the relevant particles and physics processes, and the initial event kinematics. A manual for the application programmer is found in the User's Guide: For Application Developers .

Using standard components of Geant4, a **framework provider** would add interfaces to external tools, such as for example, to Computer Aided Design (CAD) programs, Object-Oriented Data Base Management Systems (ODBMS) and graphics systems. This requires the development of new classes overloading standard Geant4 functionality and hence a solid understanding of object-oriented Programming. A manual for the framework provider is found in the User's Guide: For Toolkit Developers .

References

All user documentation can be found on the Geant4 homepage http://cern.ch/geant4 .

References for Object-Oriented Technology:

- Grady Booch, Object-Oriented Analysis and Design with Applications The Benjamin/Cummings Publishing Co. Inc, 1994, ISBN 0-8053-5340-2
- R.C.Martin, Designing Object-Oriented C++ Applications Using The Booch Method, Prentice Hall 1995, ISBN 0-13-203837-4;
- E. Gamma, et al., Design Patterns - Elements of Reusable Object-Oriented Software, Addison Wesley 1995, ISBN 0-201-63361-2;

References for C++:

- B.Stroustrup, C++ Programming Language 3rd Edition, Addison Wesley, ISBN: 0-201-88954-4
- I.Pohl, Object-Oriented Programming Using C++, 2nd Edition, Addison Wesley, ISBN: 0-201-89550-1.

Chapter 6. Computing Environment Required by the Geant4 Toolkit

The Geant4 toolkit is available for a variety of operating systems:

- flavors of UNIX,
- Linux,
- and Windows systems.

In order to link and build the program only two underlying software packages are mandatory:

- CLHEP (Class Library of High Energy Physics) and the
- STL (Standard Template Library for fundamental classes like C++ containers and strings).

The Geant4 source code is available from the Geant4 web pages while CLHEP is available from the CLHEP Home Page . For details on setting up the computing environment see the Installation Guide .