Use of GEANT4 in CMS

The OSCAR project

P. Arce[1], *S. Banerjee*[2], *S. Giani*[3], *R. Gokieli*[3], *I. González*[1,3], *J. Klem*[3,4], *M. Kossov*[5], *V. Lefébure*[3], *A. Nikitenko*[6], *M. Schröder*[3], *J.P. Wellisch*[3] *for the CMS software team*

[1] Instituto de Física de Cantabria, Univ. de Cantabria - CSIC, Santander, Spain
[2] Tata Institute of Fundamental Research - EHEP, Mumbai, India
[3] CERN, European Laboratory for Particle Physics, Geneva, Switzerland
[4] Helsinki Institute of Physics, Helsinki, Finland
[5] ITEP, Moscow, Russia
[6] HIP, Helsinki, Finland, and on leave from ITEP, Moscow, Russia

Abstract

The CMS experiment has decided to use the GEANT4 toolkit for the detector simulation. We will show how we will do the transition from the GEANT3.21 based simulation to the GEANT4 based simulation. We will describe the design of the CMS geometry and hit classes and the status of the implementation. We will also report on the experience made with GEANT4 navigation and tracking using that geometry description. We have also implemented the geometry of some test-beam setups.

Keywords: geant, simulation, cms, geometry

1 Introduction

GEANT4[1] is a toolkit for detailed as well as for fast Monte Carlo simulation of High Energy Physics experiments. This package will substitute its predecessor GEANT3.21[2] as the basis for the CMS[3] detector simulation in an Object Oriented environment. Enhanced capabilities and performance are expected from this new package (GEANT4). Although GEANT4 is based on the experience acquired for GEANT3.21, it completely remodels the physics processes and improves the precision in prediction. The transition between a GEANT3.21 based simulation package to the GEANT4 based one is explained in section 2.

An adequate implementation of the CMS geometry and the ability to produce hits are the first steps towards a complete simulation program. We have performed a first iteration in that direction using GEANT4. Our work lead us to the definition of the model described in section 3.

The Unified Modelling Language (UML)[4] is used in this paper to draw the class and category diagrams.

2 GEANT3.21 to GEANT4 transition

CMSIM is the current CMS software for simulation based on GEANT3.21 and FORTRAN77. Its development has been stopped and it will only be modified for maintenance purpose. In the meantime, CMS has been involved in the development of GEANT4 since the beginning. A new simulation software, called OSCAR[1], that uses GEANT4 is currently being built in C++. OSCAR will write the simulated hits (and simulated digits for some sub-detectors) into an OO database. The detector response (digitisation) is luminosity-dependent because of the pile-up. Therefore, the reconstruction program (ORCA[2][5]) has the responsibility to simulate the digitisation. It will also

[1] **O**bject oriented **S**imulation for **C**MS **A**nalysis and **R**econstruction
[2] **O**bject oriented **R**econstruction for **C**MS **A**nalysis

reconstruct physical objects (particles, jets, tracks) both on the trigger and at the off-line analysis level.

Several milestones have been defined within the collaboration for OSCAR project in order to make sure that a final reliable GEANT4 based simulation is achieved:

- Proof of Concept, due Summer 1998.
- Functional Prototype, due December 1999.
- Fully Functional Prototype, due June 2001.
- Production Version, due December 2003.

The first one was reached before summer 1998. CMS is now working towards the completion of the functional prototype which is expected to be finished in a short period.

3 CMS Geometry Model using GEANT4

Rather than following the physical division of CMS into sub-detectors, the code was structured into categories depending on the different responsibilities (see figure 1). Besides, we adopted the strategy of having a unique description of the geometry, completely independent of the different views that different applications (simulation, reconstruction, visualisation, ...) may need. In this approach a set of core factory classes define CMS. The simulation classes are able to extract from them the needed information to construct the GEANT4 geometry[3]. More information can be found in [6].

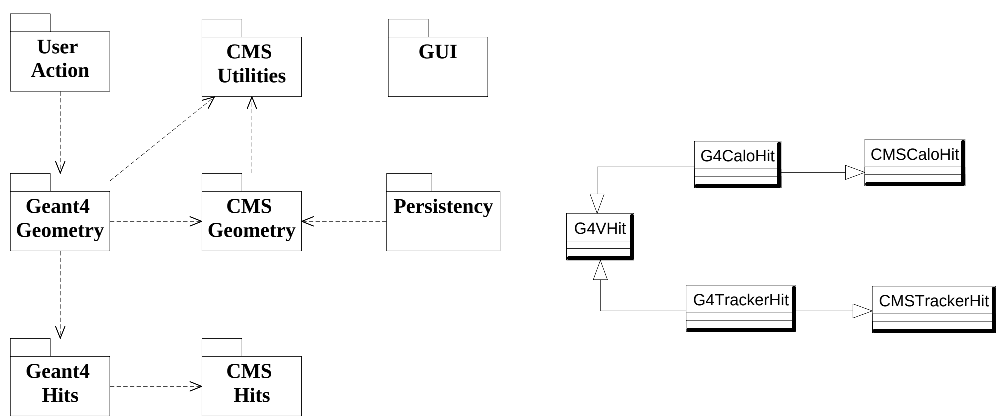

Figure 1: *OSCAR category diagram (left) and hits class diagram (right)*

3.1 The Geometry classes

There are three categories that deal with the geometry representation: CMS geometry, GEANT4 geometry and Persistent geometry. In all of them a polymorphic structure is used in the class diagrams. In each of these categories a base class maps the features expected in all the sub-detectors and handles the physical hierarchy of sub-detectors. Each sub-detector specialises in turn the base class with the specific implementation of its behaviour.

The way in which each CMS sub-detector geometry class stores the information is left free to the developer, so that he/she optimises it using the most convenient approach for his/her specific

[3] This model allows to easily extend the system to, e.g., new simulation or reconstruction, packages without affecting the core classes.

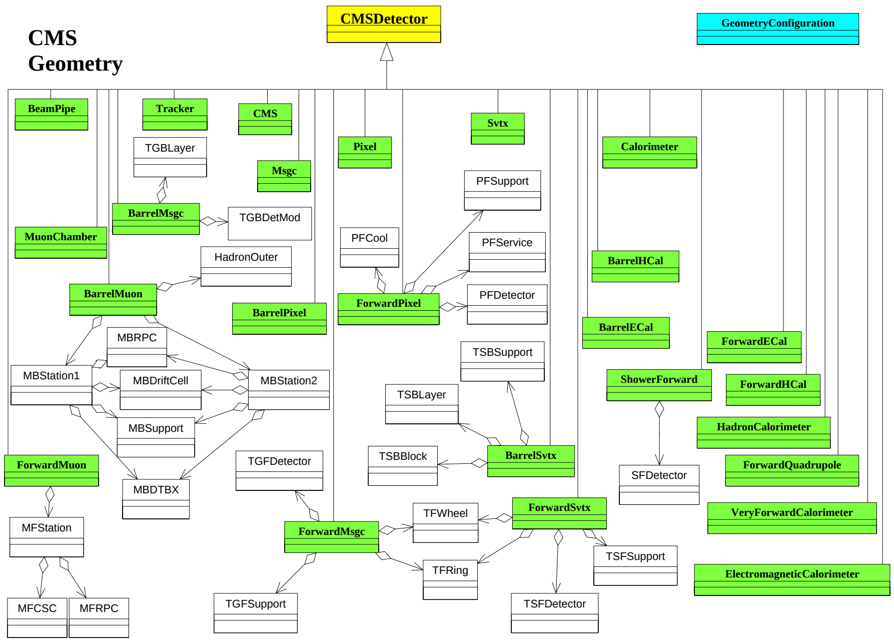

Figure 2: *CMS geometry class diagram. The base class* `CMSDetector` *is printed in yellow (light grey), while the main sub-detectors classes are in green (medium grey). The class in blue (medium grey at the upper right corner) takes care of the configuration of the geometry.*

choice. On the other hand, the handling of the detector hierarchy is implemented in the base class. The class diagram for this category is drawn in figure 2.

There is one class per sub-detector in the `GEANT4` geometry category. Each detector class in this category inherits from a base class mapping the general behaviour expected from `GEANT4` and from the corresponding counterpart in the CMS geometry category. The use of double inheritance instead of single inheritance plus composition was motivated by the fact that this classes are thought as extensions of the CMS ones. Besides, the handling of the hierarchy and the implementation of these classes becomes easier (no need to use a tell/set method per parameter). The design of the base class helps a modular approach and easy geometry interchanging at the level of sub-detectors, allowing an easy transition from the simulation of the entire CMS detector to that of just a part of it, or even to a test-beam geometry. The *sensitive* property of each single detector as well as its visualisation attributes can be selected at run time. An example of the visualisation can be seen in figure 3.

The construction of the geometry starts from the top mother detector (i.e. CMS) and is automatically propagated to its constituents (e.g. Beam-Pipe, Tracker, Calorimeters, ...) which recursively act as mother volumes.

The persistent model for the geometry uses Objectivity and it is not yet fully implemented.

3.2 Hits and Sensitive Detectors

CMS has identified two major types of detectors: the tracker-like ones and the calorimeter-like ones. Therefore a group of classes (sensitive detector, hit and hits collection) has been implemented for each type. As in the geometry case, the specific `GEANT4` features for the hit classes have been detached from the base classes which only implement the CMS behaviour. The `GEANT4` specific functionality is coded into the new classes inheriting from the CMS hit classes and from

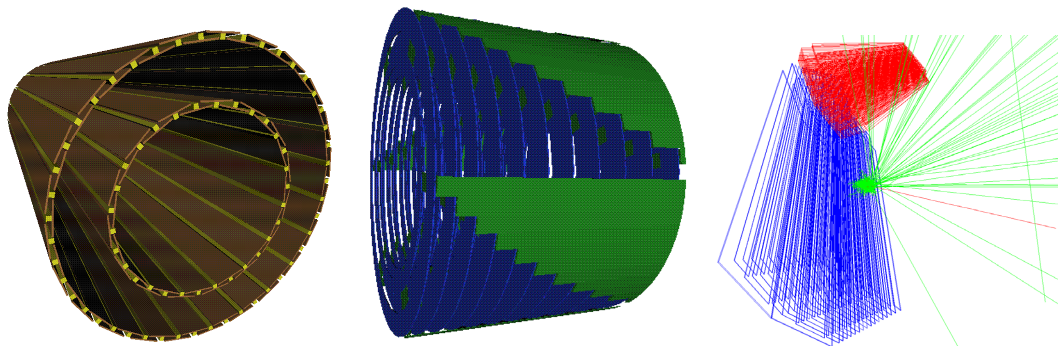

Figure 3: *Barrel Pixel (left), Forward MSGC (middle) and 99 HCAL test-beam with a 20 GeV electron (right).*

a GEANT4 abstract base class (see figure 1). The sensitive detectors require the identification of sensitive units. A convenient framework is provided with a base class that needs to be specialised for each sub-detector with sensitive units.

3.3 User Interaction

GEANT4 permits (and sometimes requires) the implementation of several classes that can extend its capabilities or that allow to interact with the normal flux of the program. CMS has implemented all these classes:

- Detector construction: Defines how the geometry construction is to be done.
- Physics list: Specifies the particles and processes to be used.
- Primary generator: Allows to select a default particle type, direction and energy. The default particle behaviour in GEANT4 has been extended with a random direction option and the possibility to read events from PYTHIA.
- Run, Event, Tracking and Stepping: They provide a way to plug-in user-defined actions in the middle of a run, an event, during tracking or at each step. CMS has extended this model further through the concept of *action units* objects which can be assigned to each of the action classes and that define a basic action to be performed. Therefore a group of different action units can be provided to the user allowing him to select a set of them to be easily plugged into the code.

3.4 Utilities

To allow defining materials in ways not directly supported by GEANT4 and to avoid multiple instances of the same material, CMS implemented a factory of materials. The same approach was taken for the most commonly used rotation matrices, for the same reasons.

GEANT4 allows to simulate the particles inside any field. A three dimensional description of the CMS magnetic field has been implemented using a 2-D field map and can be activated in the context of tracking inside the CMS detector. A class completely independent from GEANT4 implements this, and it may be reused in any other application that may need it. A new class specialises it by inheriting from it and from the virtual base class provided by the GEANT4 toolkit.

4 Conclusions

CMS is on the way towards a functional prototype of its GEANT4 based simulation. A model for the geometry has been implemented which is both extensible and flexible. This model has been succesfully exported to other areas such as the definition of hits, materials and rotations. We have also extended the GEANT4 basic functionality and adapted it to the CMS needs. The current status allows tracking through the detector and the recording of hits from most of the detectors for histogramming purposes. We have also started the comparison between the GEANT4 based simulation of detector components and sub-detectors with test-beam data and GEANT3.21 based simulations.

References

1 S. Giani *et al.*, "GEANT4: An Object-Oriented Toolkit for Simulation in HEP", CERN LHCC/98-44
`http://wwwinfo.cern.ch/asd/geant/geant4.html`

2 Application Software Group, CN Div., "GEANT Detector Description and Simulation Tool (Version 3.21)", CERN Program Library W5013.

3 CMS Collaboration, "CMS Technical Proposal", CERN/LHCC 94-38, LHCC/P1, 15 December 1994

4 `http://www.rational.com/uml/`

5 CMS Collaboration, "The Design, Implementation and Deployment of a Functional Prototype OO Reconstruction Software for CMS. The ORCA project", to be presented by V. Innocente and D. Stickland in this conference, 7-11 February 1999

6 S. Banerjee, I. González, V. Lefébure "CMS simulation software using Geant4", CMS Note 1999/072, 10 December 1999.