The ALICE Simulation Framework

I. Hrivnacova[1] *and A. Morsch*[2]
For the ALICE Off-line Project - ALICE Collaboration

[1] Nuclear Physics Institute of Academy of Sciences of Czech Republic, Rez, Czech Republic
[2] CERN, Geneva, Switzerland

Abstract

ALICE uses an object-oriented framework for simulation and reconstruction (*AliRoot*) based on ROOT. Here, we describe those components of the class design that represent common concepts of simulation such as particle generation and transport, detector response and detector segmentation. The smooth way of transition from existing GEANT 3 simulation software to GEANT 4 adopted by ALICE is explained. As a client of the Monte Carlo interface (pure abstract class) *AliRoot* is used both for GEANT 3 and GEANT 4 simulations. Applications for fast and slow physics and detector performance simulations as well as the relevance of the design for visualization and the link of simulation to reconstruction are discussed.

Keywords: ALICE, CHEP, simulation, OO, ROOT,

1 Introduction

ALICE (A Large Ion Collider Experiment) [1] is the only detector dedicated to the study of nucleus-nucleus interactions at the LHC. It will investigate the physics of strongly interacting matter at extreme energy densities and is designed to cover the full spectrum of hadronic and leptonic signals expected at the LHC allowing to study the phase transition from hadronic matter to deconfined partonic matter, the *quark gluon plasma (QGP)*. ALICE will be ready for data taking from the first year of LHC running expected for 2005.

The variety of physics signals and the special experimental conditions prevailing in heavy ion collisions (up to 8000 particles per unit of rapidity) had to be considered for the development of the ALICE simulation framework. It should allow to assess both, the studies of the detailed response and performance of individual sub-detector systems and the overall physics performance.

Heavy ion collisions produce events of unpreceded complexity. The spectrum of physics analysis reaches from the study of global event features for which in the order of 10000 particles have to be fully reconstructed to the study of single particles from rare processes in such events. To decrease computing time, fast simulation methods for signals and backgrounds based on the results of detailed simulations have to be applied and are ideally part of the simulation framework.

To provide the user with an easy way to study different physics processes in heavy ion collisions, but also with tools for testing and background studies the simulation framework should provide a dedicated interface to primary particle generators.

ALICE decided to integrate both simulation and reconstruction in the same object oriented framework (*AliRoot*) [2] based on ROOT [3]. The framework is successfully used during the current phase of studies for the Technical Design Reports of the individual detector systems. In the near future it has to develop to a tool for the study the overall physics performance of the ALICE detector.

2 Simulation Strategy

Starting out the development of the simulation framework for ALICE a clear distinction between immediate and long term requirements was made. It was clear, that one should profit from object oriented design as soon as possible. On the other hand, simulation tools had to be provided to the collaboration almost immediately for

- Technical Design Report studies
- Detector design optimization
- Proof of principle for new physics analysis ideas
- Integration of new detector components.

Hence, the integration of existing GEANT 3 [4] based simulation code in the framework was mandatory and the transition to GEANT 4 [5] based simulations within the same framework has been defined as a long term goal. This strategy allowed a smooth and successful migration of FORTRAN programmers and GEANT 3 users to OO (see contribution of F. Carminati to this conference).

To assure the quality of the simulations it is seen as necessary that comparison of GEANT 3 and GEANT 4 simulations can be performed using the same geometry description and output data structures. The transition to GEANT 4 can be smooth when a maximum of GEANT 3 based user code is reused. These requirements have lead to the development of a Monte Carlo interface (*AliMC*), that allows to use the same detector simulation code with different transport codes.

It has been soon realized that the same concept of modularity based on inheritance can be also used for the parts of the simulation code which are not related to particle transport such as:

- Primary particle generation
- Detector segmentation description and behavior
- Detector response

Besides allowing for a maximum of reuse of code this concept helps to assure the coherence of the simulation process comprising event generation, particle transport, signal generation, digitization, visualization and fast simulation.

As pointed out in the introduction, fast simulation methods without full particle transport are essential in the study of the physics performance of the ALICE detector. The development of a simulation run base class (*AliRun*) will allow for a coherent integration of both fast and slow simulations into *AliRoot*.

3 Monte Carlo Interface

The Monte Carlo interface has been developed as a generalization of GEANT 3 functions for definition of the simulation task. It is an abstract class that provides methods for geometry description and physics setup definition, methods for access to tracking and particle properties during stepping and methods for visualization. While the implementation of this interface for GEANT 3 was straightforward, as *AliMC* was born from GEANT 3, its implementation for GEANT 4 needs more effort.

3.1 Implementation of the Monte Carlo interface for GEANT 4

The GEANT 4 Monte Carlo class (*TGeant4*) is implemented in terms of five manager classes: geometry, physics, step, run and visualization manager. This decomposition of functionality respects the hierarchy of the categories in the design of GEANT 4.

The geometry manager class has been designed as a client of the *g3tog4* package, the stand-alone package provided by GEANT 4 for automatic conversion of GEANT 3 geometry to GEANT 4. The development of this class has resulted to our contribution to the *g3tog4* package in the last GEANT 4 release.

The main difference between the standard usage of *g3tog4* package and its usage in ALICE is that the former starts from the GEANT 3 output file in ZEBRA format that is transformed into ASCII data file which is afterwards read and processed by *g3tog4* methods while the latter starts from the *AliRoot* detector classes that are clients of the *AliMC* interface. Generating of the ASCII data files with geometry description from *AliRoot* is possible, too, namely for debugging reasons. The generated data files can be read back in *AliRoot* and they can be processed by the standard *g3tog4* tool as well.

The geometry manager, as well as *g3tog4* tool, supports almost all GEANT 3 options for geometry definition: passing parameters from mother volume to its daughters is enabled by temporary tables corresponding to GEANT 3 volumes definition, divided volumes are supported for most of the volume shapes and are represented by replicated physical volumes in GEANT 4. The 'MANY' option that introduces boolean operations on shapes in GEANT 3 geometry is not supported - the support for particular cases is being considered at present time.

The physics manager class provides the GEANT 4 physics list construction starting from the GEANT 3 cuts and physics process control parameters. The tracking media parameters that apply the cuts and physics process control parameters to particular volumes in GEANT 3 are applied to GEANT 4 logical volumes with usage of the user limits (*G4UserLimits*) derived class and two special processes: special cuts and special flags (process control) processes (derived from *G4VProcess*). This part of the *AliMC* interface is under development and only a subset of GEANT 3 parameters is supported now.

The step manager class works as an adapter between the *AliMC* interface and GEANT 4 step manager (*G4StepManager*), it provides access to properties of the tracking particle during stepping and is already completely implemented. The visualization manager class is designed to adapt GEANT 4 visualization commands to the GEANT 3 like methods defined in *AliMC* and it is not yet implemented. The run manager class provides GEANT 4 run control for the application main program or its run manager.

The present implementation of the MC interface for GEANT 4 enables to construct the GEANT 4 geometry for the whole ALICE detector defined in *AliRoot* with exception for the subsystems that use the "MANY" option and to reuse the *AliRoot* detector response code. The simulated data (hits, digits) are stored in the ROOT output file(s) the structure of which is defined in the *AliRoot* framework and they can be directly processed with the same digitization and reconstruction code as the data from GEANT 3 simulation.

4 Generator Interface

To provide the user with an easy way to study the variety of physics processes in heavy ion collisions, but also with tools for testing and background studies *AliRoot* provides an interface to primary particle generators *AliGenerator*. The main methods of the class have the function to make the generator known to the run manager (*gAlice*), to generate the primary particles and put them on the stack, to set the allowed kinematic region and to define the vertex position and vertex smearing.

At present the following implementations exist in *AliRoot*:

- Generator using interface to external generators via the ROOT *TGenerator* interface (for example *TPythia*).
- Generators reading external event files in ROOT format.
- Generation of particles from transverse momentum and rapidity distributions given by a

function library.

- Reference event generator for background studies.
- Testing tools: particle gun, generation of particles on a 3dim grid ..
- Container class for *AliGenerator* (*AliGenCocktail*).

AliGenCocktail is a recursive implementation of *AliGenerator* in the sense that it is derived from *AliGenerator* and at the same time a container class for objects of type *AliGenerator*. It allows to build up an event from several generators. This implementation provides an appropriate tool to a typical problem in heavy ion physics: particles from many different sources can be present in the same event and have to be combined for example to obtain and analyze an invariant mass spectrum (cocktail plot). The simulation can be performed by either using full particle transport or using primary particles together with a fast detector response simulation for momentum smearing and efficiencies.

5 Segmentation Interface

The detector segmentation interface *AliMUONSegmentation* was developed to provide an interface that reflects common 'behavior' of detector segmentation such as

- Pad to real coordinate transformation and vice versa.
- Iteration over pads.
- Providing next neighbors of a given pad.

The interface is particularly adapted to the simulation of the tracking and trigger chambers of the ALICE muon spectrometer which have an irregular segmentation, i.e. cell size varying with the radial distance to the beam but is also used for the simulation of the ALICE Ring Imaging Cherenkov Counter. Instead of presenting implementation details we want to stress the fact that a wide range of clients for such an interface have been identified and are used within *AliRoot* for *simulation and reconstruction*. Among them are:

- Signal generation: Spreading charge over a pad region.
- Visualization of hits together with resulting clusters.
- *Neighbors(..)* method is used for a recursive cluster finding.
- Hit reconstruction from clusters.

6 Conclusion

ALICE uses a ROOT based OO framework for simulation and reconstruction (*AliRoot*). Interface classes are important building blocks of the framework providing modularity and coherence of the simulation process.

The *AliMC* Monte Carlo interface allows integration of existing GEANT 3 based simulation code and in a later stage a transition from GEANT 3 to GEANT 4 with reuse of the existing code for the geometry description and detector simulation.

References

1 ALICE Collaboration, Technical Proposal, CERN/LHCC 95-71.
2 http://alisoft.cern.ch/offline
3 http://root.cern.ch
4 http://wwwinfo.cern.ch/asd/geant
5 http://wwwinfo.cern.ch/asd/geant4/geant4.html