Tools for simulation and analysis

A. Pfeiffer[a], M. G. Pia[b] (on behalf of the Anaphe and Geant4 Collaborations)

[a]CERN, CH-1123 Geneva, Switzerland

[b]INFN Sezione di Genova, Via Dodecaneso 33, 16146 Genova, Italy

Software plays an incrwasingly important role in the design and operation of large-scale HEP detectors, as well as in the analysis of experimental data. In the recent past, new software technologies have emerged in the HEP domain, and new software systems, in particular for detector simulation (Geant4) and data analysis (Anaphe), have been developed to address the challenging requirements of new-generation HEP experiments.

1. The challenge of HEP software

The software for high energy physics experiments faces an unprecedented challenge, determined by the complexity of the physics under study, of the detectors and of the environments where they operate.

The complexity of the physics of the LHC experiments, related to the high machine luminosity and the significant QCD background, is the source of their enormous computational requirements: 5-8 Petabytes of data will be generated each year, and their analysis will require ≈ 10 Petabytes of disk storage and the equivalent of 200,000 of today's fastest PC processors.

The software for astroparticle experiments, which embrace a wide spectrum of scientific programmes, from underground neutrino physics and dark matter searches to astrophysics measurements on satellites to cosmic ray physics, is characterised by a similar variety of requirements. The physics models necessary for the simulation of such experiments, also accounting for the space radiation environment or underground backgrounds, span an energy range from the order of few eV to the PeV scale. Moreover, for some of these experiments, like, for instance, those on satellites, simulation is often mission critical, thus requiring high reliability and rigorous software engineering standards in addition to powerful functionality.

While facing the diversity of requirements from many different applications, the software for HEP has to deal with a growing trend to share functionalities across diverse fields: for instance, models to extend the simulation of electromagnetic interactions down to low energies ($< 1keV$), motivated originally from medical physics applications, have been successfully applied to astroparticle experiments, and are of interest for the precise simulation of LHC detectors.

The intrinsic complexity of the software for the new generation of HEP experiments is further complicated by the rapidly changing computing environment. The evolution of hardware, operating systems and software over the long time scale of LHC operation, as well as likely changes in the requirements of the experiments, imply that the software being built now must be able to anticipate changes.

The response to such challenges consists of the adoption of a rigorous software engineering, together with a collaborative effort to share resources effectively. The adoption of object oriented technology makes the software open to extension and evolution, and facilitates its maintenance. A component-based architecture allows both the specialisation of components and their easy integration with experimental frameworks, providing the flexibility to customise the software by means of alternative, interchangeable implementations. Computing facilities implemented as a global computational grid aim to integrate geographically distributed computing fabrics into a worldwide virtual computing environment.

2. The Geant4 Simulation Toolkit

Geant4 [1] is an object oriented simulation toolkit designed for the new generation of high energy physics experiments; its current applications also cover astrophysics, nuclear physics, medical physics, space science and radiation background studies.

Software engineering plays a fundamental role in Geant4: advanced techniques are adopted in response to the requirements of functionality, modularity, extensibility and openness, resulting in a wide set of capabilities to model detectors and particle interactions with matter over a wide energy range.

Thanks to its modular architecture, Geant4 can be easily integrated into the complex software frameworks of large-scale HEP experiments, to collaborate with other software components such as data analysis.

2.1. Overview of Geant4 functionality

The Geant4 kernel handles the management of runs, events and tracking. The Event package provides an abstract interface to external physics event generators for the creation of the primary particles. Geant4 Tracking handles the propagation of a track, determined by physics interactions. It is completely general and common to all processes for any particle type, thus allowing great flexibility in the implementation of a variety of physics processes, as well as openness to further extensions. The management of particles is based on Particle Data Group compliant definitions and data, including their decay processes and modes.

The Geant4 Geometry package provides tools to describe the detailed geometrical structure of a detector; it also handles the equation of motion solvers in different fields and geometrical boundary conditions for the propagation of particles. An ISO STEP compliant solid modeller allows the exchange of models from CAD systems. Multiple solid representations, such as Constructive Solid Geometry or Boundary Represented Solids are available. Boolean operations on solids are supported.

The Geant4 Materials package allows the description materials consisting of a single element or a composition of elements, which in turn can consist of a single isotope or a mixture of isotopes.

The Hits and Digi domains provide the functionality to reproduce the readout structure of a detector and its electronic response, independently from the geometry used for tracking particles.

Geant4 Visualisation provides the capability to visualise detector geometry, particle trajectories, tracking steps, hits and text. Its design, based on abstract interfaces, makes Geant4 independent from any particular graphics system; at the same time it allows multiple implementations of drivers to interface the simulation with a variety of such systems. The User Interface domain adopts a similar approach, allowing the usage of a variety of user interfaces, from simple command-line driven ones to sophisticated GUIs.

A Fast Parameterisation facility is integrated with the full simulation, allowing independent and simplified detector descriptions and direct production of hits.

Extensive possibilities for interaction with the Geant4 system are offered to the user via a set of dedicated user-action classes.

2.2. Geant4 physics

One of the most important goals of Geant4 consists in making the design and implementation of the physics open and transparent. The Geant4 design makes physics data, models and assumptions transparently accessible to the user, rather than hard-coded in black-box packages, thus improving the verification and the reliability of simulation results. It also exposes the granular implementation of the physics, each component of which can be inspected at source code level. Thanks to object oriented technology, a variety of alternative or complementary physics models can be provided for the same physical process, with openness to further extensions.

Geant4 electromagnetic physics [2] provides a variety of implementations of electron, positron, photon, charged hadron and ion interactions. Photon processes include Compton and Rayleigh scattering, γ-conversion and the photoelectric effect. Electron/positron processes include bremsstrahlung, ionisation and δ-ray production, positron annihilation and synchrotron radiation. The ionisation and energy loss are available for hadrons and ions as well. The multiple scattering process handles all charged particles, computing the mean path length correction and the mean lateral displacement. Low energy extensions [3] are implemented, down to 250 eV for photons and electrons, and to the energy corresponding to the ionisation potential of the material for hadrons and ions. The validity range of all the muon processes, based on theoretical models, scales up to the thousand PeV region, allowing the simulation of ultra-high energy and cosmic ray physics.

Geant4 can also handle the optics of scintillation and Cherenkov detectors and their associated light guides, including a set of dedicated processes for optical photons: refraction and reflection at medium boundaries, absorption and Rayleigh scattering.

The Geant4 Toolkit provides a rich set of physics models for the simulation of hadronic interactions [4]. The models follow three different basic approaches: data-driven, parameterisation-driven and theory-driven modelling; in the overall framework they offers both complementary and alternative options.

3. Analysis tools

3.1. Abstract Interfaces for Data Analysis

The AIDA [5] project represents an ongoing effort to define a set of common abstract interfaces for data analysis in high energy physics experiments; it encompasses histograms, ntuples, functions, vectors, fitting, plotting. The interfaces have been defined in C++ and Java.

Abstract interfaces, i.e. classes with pure virtual methods and only inheriting from other abstract interfaces, enforce a weak coupling among the components of a software system: the components may therefore evolve independently, thus reducing the maintenance overhead. This approach also provides a great flexibility, since different implementations of a component can be easily interchanged by using dynamically loadable libraries.

3.2. The Anaphe Analysis Toolkit

The Anaphe [6] toolkit provides an object oriented software environment for data analysis in high energy physics experiments, compliant with AIDA interfaces. The architecture of Anaphe is the result of a significant effort to minimise the dependencies: the system is characterised by a strict partitioning into independent packages, which communicate only through interfaces. This component approach facilitates the maintainability of the software, as well as the flexibility of customisation of user applications.

A range of commercial and public domain libraries is used to cover basic functionalities; on top of these libraries a set of HEP-specific class libraries for histogram management, fitting, plotting and ntuple-like data analysis has been developed. In some cases alternative implementations for the same interface are provided.

An interactive analysis and visualisation tool (Lizard) provides a flexible Python-based framework, which can be fully configured at run-time to load any of the Anaphe libraries or external modules.

REFERENCES

1. S. Agostinelli et al., GEANT4: a simulation toolkit, SLAC-PUB-9350.
2. S. Chauvie et al., Geant4 electromagnetic physics, Proc. of the Monte Carlo 2000 Conference, Springer, 2000.
3. S. Chauvie et al., Geant4 low energy electromagnetic physics, Proceedings of CHEP2001, Beijing, 2001.
4. J.P. Wellisch, Hadronic shower models in Geant4: the frameworks, Comput.Phys. Commun. No. 140 (2001) 65.
5. G. Barrand et al., Abstract Interfaces for Data Analysis: Component Architecture for Data Analysis Tools, Proceedings of CHEP2001, Beijing, 2001.
6. B. Ferrero Merlino et al., Anaphe: OO Libraries and Tools for Data Analysis, Proceedings of CHEP2001, Beijing, 2001.