Available online at www.sciencedirect.com
SCIENCE DIRECT®
Nuclear Physics B (Proc. Suppl.) 150 (2006) 62–65
NUCLEAR PHYSICS B PROCEEDINGS SUPPLEMENTS
www.elsevierphysics.com

GLAST LAT Full Simulation

Luca Baldini[a], Denis Bastieri[b], Praveen Boinee[cd], Monica Brigida[e], Giuseppe Cabras[cd], Claudia Cecchi[f], Alessandro De Angelis[cd], Dario Favretto[cg], Massimo Fiorucci[f], Marco Frailis[cd], Fabio Gargano[e], Riccardo Giannitrapani[cd], Nicola Giglietto[e], Michael Kuss[a], Luca Latronico[a], Andrea Lionetto[h], Francesco Longo[di], Francesco Loparco[e], Pasquale Lubrano[f], Francesca Marcucci[f], Mario Nicola Mazziotta[e], Edoardo Milotti[cd], Aldo Morselli[h], Nicola Omodei[aj]*, Monica Pepe[f], Riccardo Rando[b], Massimiliano Razzano[a], Gloria Spandre[a] and Gino Tosti[f]

[a]INFN Pisa, Italy

[b]Università di Padova and INFN Padova, Italy

[c]Università di Udine, Italy

[d]INFN Trieste, Italy

[e]Università di Bari and INFN Bari, Italy

[f]Università di Perugia and INFN Perugia, Italy

[g]CERN, Geneva, Switzerland

[h]Università di Roma "Tor Vergata" and INFN Roma-2, Italy

[i]Università di Trieste, Italy

[j]Università di Siena, Italy

This paper presents the simulation of the GLAST high energy gamma-ray telescope. The simulation package, written in C++, is based on the Geant4 toolkit, and it is integrated into a general framework used to process events. A detailed simulation of the electronic signals inside Silicon detectors has been provided and it is used for the particle tracking, which is handled by a dedicated software. A unique repository for the geometrical description of the detector has been realized using the XML language and a C++ library to access this information has been designed and implemented. As first application of the GLAST LAT software, one day of simulated data has been produced. This paper outlines the contribution developed by the Italian GLAST software group.

1. Introduction

The Gamma-ray Large Area Space Telescope (GLAST) is an international mission that will study the high-energy phenomena in gamma-rays universe [1]. GLAST is scheduled for launch in 2007.

GLAST is instrumented with a hodoscope of Silicon planes with slabs of converter, followed by a calorimeter; the hodoscope is surrounded by an anticoincidence (ACD). This instrument, called the Large Area Telescope LAT, is sensitive to gamma rays in the energy range between 30 MeV and 300 GeV. The energy range, the field of view and the angular resolution of the GLAST LAT are vastly improved in comparison with those of its predecessor EGRET (operating in 1991-2000), so that the LAT will provide a factor of 30 or more advance in sensitivity. This improvement should enable the detection of several thousands of new high-energy sources and allow the study of gamma-ray bursts and other transients, the resolution of the gamma-ray sky and diffuse emission,

*Corresponding author: nicola.omodei@pi.infn.it

0920-5632/$ – see front matter
doi:10.1016/j.nuclphysbps.2005.04.062

the search for evidence of dark matter and the detection of AGNs, pulsars and SNRs. A detailed description of the scientific goals of GLAST mission and an introduction to the experiment can be found in [2].

GLAST is a complex system, and detailed computer simulations are required to design the instrument, to construct the response function and to predict the background in the orbit. To accomplish these tasks an object-oriented C++ application called *Gleam* (GLAST LAT Event Analysis Machine) was adopted and implemented by the GLAST LAT collaboration. A brief description of the its structure can be found in [3].

The GLAST off-line software is based manly on Gaudi, a C++ framework, originally developed at CERN [4]. In the GLAST software, Gaudi manages the loop of particles to be simulated, then a series of algorithms are applied to each of them to get the result of the complete simulation and reconstruction chain. The structure of the GLAST off-line software is described in figure 1. An important characteristics is the separation of the packages according to their responsibilities. Most packages has been developed by the GLAST collaboration for the specific items required by the simulation of a high energy gamma-ray telescope. The main components of Gleam developed by our group are the subject of this paper.

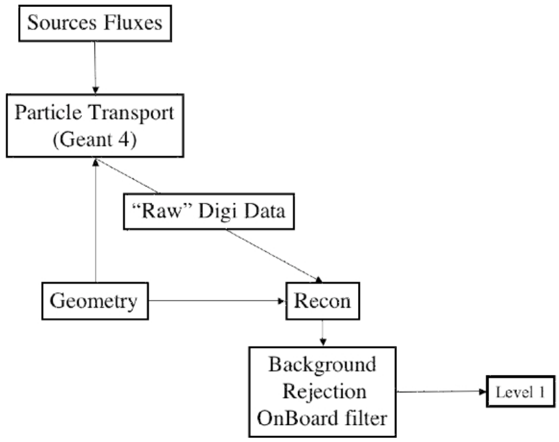

Figure 1. General scheme for simulation and reconstruction within the GLAST off-line software framework

2. The Source Generation package

The Source Generation is the first algorithm called within the particle loop. Its task is to generate particles according to certain characteristics. This algorithm must store the information on the temporal and spectral behaviour of the source, as well as on the orbital characteristics of GLAST. An extension of this algorithm has been implemented for simulating transient sources such as Gamma-Ray Bursts (GRB). It can be used for studying the capability of GLAST for the observation of rapid transient fluxes in general. The physics adopted is based on the fireball model of Gamma Ray Burst, for which a series of shells is injected in the circumburst medium with different Lorentz factor [5]. When a faster shell reaches a slower one a shock occours, and an accelerated electron distribution is obtained due to the shock acceleration process. Some of the energy dissipated during the shock is converted into a randomly oriented magnetic field. The electrons can loose their energy via synchrotron emission. The caratheristic synchrotron spectrum is boosted (and beamed) thanks to the Lorentz factor of the emitting material. The higher energy part of a GRB spectrum can be obtained keeping into account the possibility of Compton scattering of the synchrotron photons against the electron accelerated by the shock (Self Synchrotron Compton)[6].

3. The Simulation package

The algorithm which is responsible for generating the interactions of particles with the detector is based on the Geant4 MonteCarlo toolkit [7] which is an Object Oriented (OO) simulator of the passage of particles through matter. Its application areas include high energy physics and nuclear experiments, medical science, accelerator and space physics.

Geant4 (G4) provides a complete set of tools for all the domains of detector simulation: Geometry, Tracking, Detector Response, Run, Event and Track management, Visualisation and User Interface. A large set of Physics Processes handle the diverse interactions of particles with matter across a wide energy range, as required by G4 multi-disciplinary nature; for many physics processes a choice of different models is available. In addition a large set of utilities, including a powerful set of random number generators, physics units and constants, a management of particles compliant with the Particle Data Group, as well as interfaces to event generators and to object persistency solutions, complete the toolkit. G4 exploits advanced Software Engineering techniques and OO technology to achieve the transparency of the physics implementation and hence provide the possibility of validating the physics results. The OO design allows the user to understand, customize or extend the toolkit in all the domains. At the same time, the modular architecture of G4 allows the user to load and use only the components needed. To build a specific application the user-physicist chooses among these options and implements code in user action classes supplied by the toolkit [8].

Within the GLAST software the simulation is managed by the Gaudi algorithm G4Generator [9]. The main simulation is controlled by a customized version of the G4 standard RunManager. Since the GLAST main event loop is driven by Gaudi and it will not use any graphics or data persistency features of Geant4, we have included in the RunManager only the real necessary parts to setup and run the generator.

A validation procedure of the electromagnetic and the hadronic physical processes relevant to GLAST is being designed. Such procedure could help in validating the data produced with the full chain of simulation, digitization, reconstruction and analysis.

4. The Digitization package

To implement a detailed digitization of the Tracker system a full simulation code has been developed. It takes into account all the main physical processes that take place in a silicon strip detector (SSD) when it is crossed by an ionizing particle [10]. The input parameters of the code are the entry and exit points of the particle in a silicon ladder and the energy deposited by the particle, provided by the simulation package. Starting from these parameters, the e-h pairs are generated along the track and are propagated towards the electrodes. The current signals induced on each strip are evaluated and are converted into voltage signals using the transfer function associated to the detector electronics, taking into account the detector noise as well as the noise associated to the electronics. The fired strips and the time over threshold (TOT) are then determined after imposing a threshold on the voltage signals. The TOT gives information about the collected charge. Our simulation of the GLAST Tracker front-end chip[10] shows that the TOT is linear with the input charge up to 50-60 fC. Laboratory tests on the front-end tracker chip confirmed the results of our simulation, in good agreement with the PSPICE[11] results.

5. The Reconstruction package

This package contains the code that reconstructs tracks from hit strips in the LAT tracker. It's organized as a series of algorithms that act successively [12]. Starting from digits generated by the Digitization package, it generates a series of clusters, that are used to find and fit the best track candidates. This last procedure is done using alternative pattern recognition algorithms and a Kalman Filter based algorithm. Finally, using the best track found, another algorithm finds the best vertex candidate for gamma events. Figure 2 shows a photon track reconstructed in the GLAST LAT together with the full MonteCarlo propagation. A slightly different approach is used to study the onboard reconstruction. The onboard reconstruction can quickly provide the reconstructed direction of the incoming photons. The main purpose in the development of such algorithm is to provide the information needed to trigger and to reconstruct the direction of fast transient sources. In the particular case of Gamma-Ray Burst the direction of the incoming photons has to be determined onboard, in order to provide in few seconds the localization

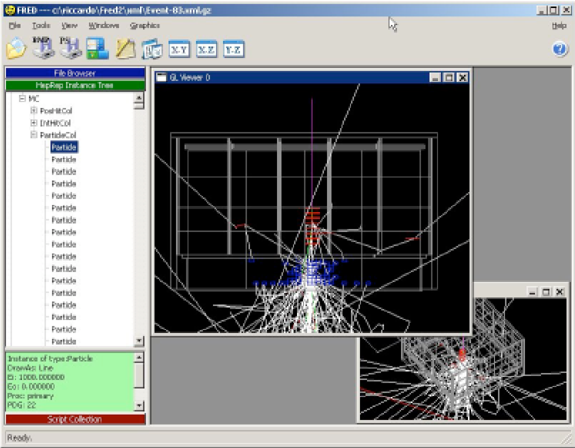

Figure 2. GLAST LAT event display based on FRED. An high energy gamma-ray interacting with the GLAST LAT detector. In the figure is visible the montecarlo propagation, the tracking algorithms and the final reconstruction of the incoming event.

of the sources.

6. Event Display and GUI

Although it is not part of the simulation, the visualization package is essential for the use of the simulation itself. A new version [13,14] of the event display based on the HepRep [15] protocol has been developed and will be integrated in the offline software as soon as possible; such a framework will allow both local event browsers directly from the GAUDI framework, serialization of the event with the use of XML and a CORBA server-client mode for remote analysis of events, with an high degree of interactivity (click and inspect) and the possibility to customize the graphical appearance of the Event. Since the HepRep protocol is completely open and transparent, it will be possible to use different graphical clients (for now WIRED[16] and FRED[14]. The figure 2 shows a recent FRED-based event display of GLAST.

7. Conclusions

The *Gleam* simulation program has been developed in the last few years and now it's ready for simulating the full GLAST satellite. *Gleam* has been used for generating one day of simulated data (Data Challenge 1) for the development of scientific analysis software. *Gleam* will be used for other MonteCarlo data taken needed for developing and testing the scientific software, and it will drive the GLAST community to the final instrumental knowledge.

REFERENCES

1. `http://glast.gsfc.nasa.gov/`
2. GLAST Science Brochure (March 2001), `http://glast.gsfc.nasa.gov/`
3. P.Boinee et al. (2003), arXiv:astro-ph/0308120
4. Gaudi: `http://proj-gaudi.web.cern.ch/`
5. T. Piran (1999), Phys. Rep., **314**, 575
6. N. Omodei and J. Cohen-Tanugi, `http://www.pi.infn.it/~omodei/GRBSpectrum/`
7. S. Agostinelli *et al.*, (2003), NIM-A, **506**, 250
8. Geant4: `http://cern.ch/geant4`
9. R. Giannitrapani *et al.*, G4Generator, GLAST internal code review (2002)
10. M. Brigida *et al.* (2002), LAT internal note, LAT-TD-1058
11. `http://www.pspice.com`
12. B. Atwood *et al.*, TkrRecon, GLAST internal code review (2002)
13. M. Frailis and R. Giannitrapani, arXiv:cs.gr/0306031. Published in eConf C0303241:MOLT010,2003
14. `http://www.fisica.uniud.it/~riccardo/research/fred`
15. J. Perl, R. Giannitrapani and M. Frailis, arXiv:cs.gr/0306059. SLAC-PUB-9908, CHEP-2003-THLT009, Jun 2003. 9pp.
16. M. Dönszelmann, A. Ballaminut, C. Colonello, E. van Herwijnen, D. Köper, J. Korhonen, M. Litmaath, J. Perl, A. Theodorou, D. Whiteson, E. Wolff', CHEP 2000, Padova, Italy, 7-11 Feb 2000