GLAST Large Area Telescope Simulation Tools

Luca Baldini[1,2], Denis Bastieri[3], Praveen Boinee[4,5], Monica Brigida[6], Giuseppe Cabras[4,5] Claudia Cecchi[7], Johann Cohen-Tanugi[2], Alessandro De Angelis[4,5], Dario Favretto[4,8], Massimo Fiorucci[7], Marco Frailis[4,5], Fabio Gargano[6], Riccardo Giannitrapani[4,5], Nicola Giglietto[6], Michael Kuss[2], Luca Latronico[1,2], Andrea Lionetto[9], Francesco Longo[5,10,†], Francesco Loparco[6], Pasquale Lubrano[7], Francesca Marcucci[7], Mario Nicola Mazziotta[6], Edoardo Milotti[4,5], Aldo Morselli[9], Nicola Omodei[2,11], Monica Pepe[7], Riccardo Rando[3], Massimiliano Razzano[1,2] Gloria Spandre[2] and Gino Tosti[7]

Abstract— **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. A new event display based on the HepRep protocol is being implemented. The GLAST satellite parameters derived from the simulation are used in a fast simulator to obtain a "snapshot" of the gamma-ray sky. This paper outlines the contribution developed by the Italian GLAST software group.**

Index Terms— **Gamma-ray satellites, Simulation, Geant4, Source modeling, Event Display, Silicon Detectors, Fast simulation.**

I. 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 2006.

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, 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++ framework called *Gleam* (GLAST LAT Event Analysis Machine) was adopted and implemented by the GLAST LAT collaboration. A brief description of the framework could 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 framework, 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 explicitely 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.

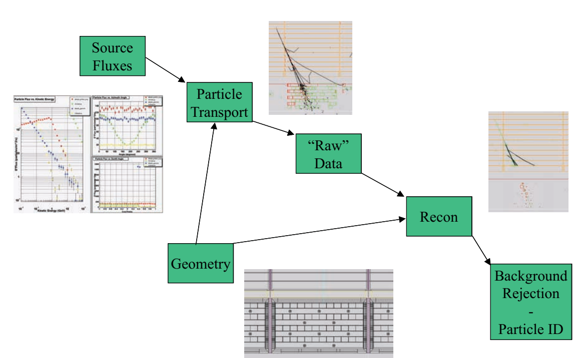

Fig. 1. General scheme for simulation and reconstruction within the GLAST off-line software framework

† Corresponding author: francesco.longo@ts.infn.it
(1) Università di Pisa, Italy
(2) INFN Pisa, Italy
(3) Università di Padova and INFN Padova, Italy
(4) Università di Udine, Italy
(5) INFN Trieste, Italy
(6) Università di Bari and INFN Bari, Italy
(7) Università di Perugia and INFN Perugia, Italy
(8) CERN, Geneva, Switzerland
(9) Università di Roma "Tor Vergata" and INFN Roma-2, Italy
(10) Università di Trieste, Italy
(11) Università di Siena, Italy

0-7803-8257-9/04/$20.00 © 2004 IEEE.

II. 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 (Inverse Compton Scattering)[6]. Figure 2 shows a light curve generated with this package.

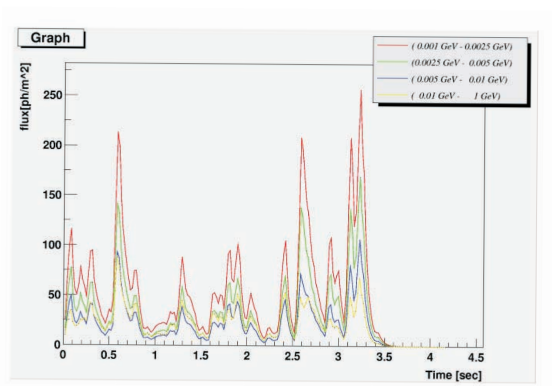

Fig. 2. An example of GRB light curve from the GLAST simulator.

III. 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, customise 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 Gleam framework 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 for setup and run the generator. RunManager itself uses the following classes:

- DetectorConstruction: this class provides the list of materials and the geometry of the detector. In our case this information is stored in XML files; to access them the DetectorConstruction class uses methods of a Gaudi service. The Geometry class implements methods to traverse the geometry of GLAST and build a concrete Geant4 representation of it; the Material class does the same for the materials definitions.
- PhysicsList: this class is the access point to the physics processes selection and customization. It uses other classes (GeneralPhysics, EMPhysics, HadronPhysics, MuonPhysics, IonPhysics) to set up particular physics processes. Since the Geant4 toolkit is open to new physics processes (along with new description of already present processes), this will be the access point for further development in the physics selection.
- PrimaryGeneratorAction: this class is in charge of production and injection of primary particles in the detector simulation. In our cases it is linked to the Gaudi algorithm that is responsable to generate the incoming fluxes of particles.
- DetectorManager: this class manages the setup and working of the sensitive detectors of the simulation and their interaction with the GAUDI Data Store. It is concretely implemented in the two subclasses PosDetectorManager and IntDetectorManager; the first one is associated with detector that saves hits information in the Si planes of the LAT tracker (TRK), while the second one is used for Anticoincidence (ACD) tiles and Calorimeter (CAL) cells.

Figure 3 shows an event generated using Geant4 within the GLAST LAT experiment.

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.

IV. 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 first version of the code has been written in FORTRAN and uses the HEED package for simulating the energy loss of charged particles in silicon. The present version of the code has been written in C++ and the process of energy loss is simulated by Geant4.

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. Figure 4 shows some results of this package, concerning the signal generated in a silicon ladder equipped with the GLAST LAT electronics.

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 (see figure 5).

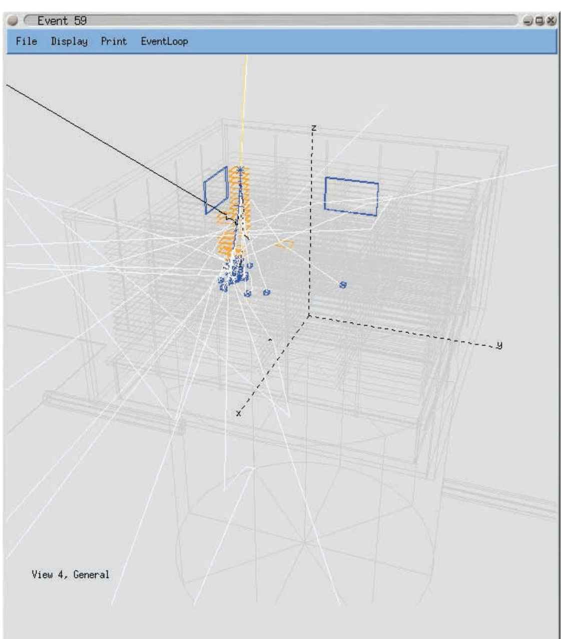

Fig. 3. High energy gamma-ray interacting with the GLAST LAT detector

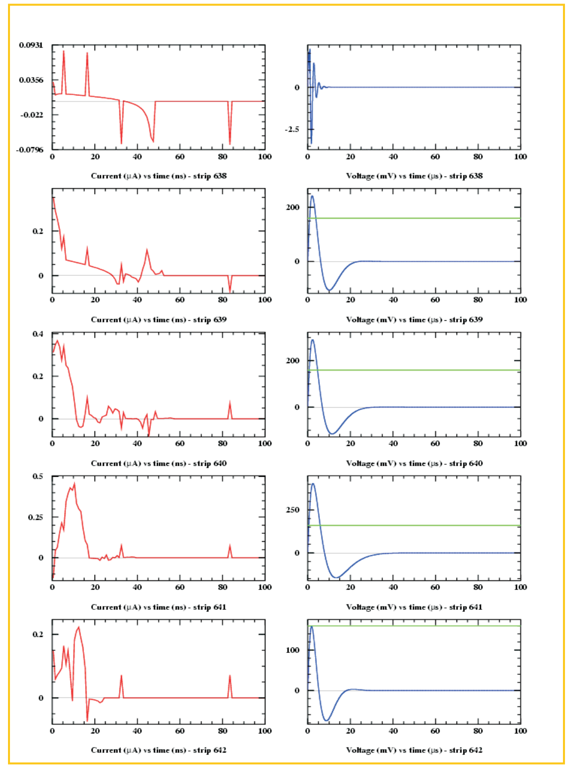

Fig. 4. Charge sharing for a 5 GeV electron, crossing the silicon wafer at large zenith angle (60°). The superimposed green line represents the readout threshold voltage.

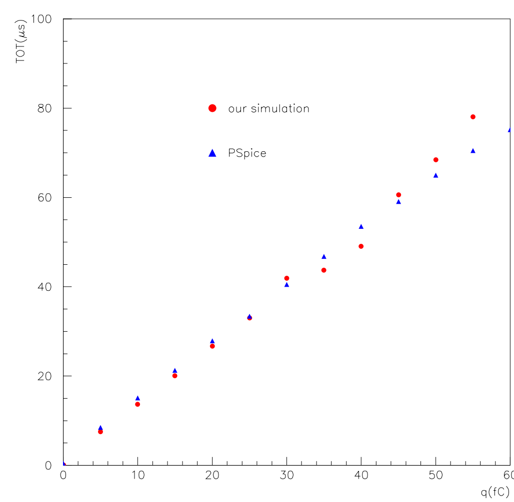

Fig. 5. Comparison of the time over threshold measured by the GLAST front-end electronics simulated with the Digitization package and with PSPICE.

V. 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 6 shows a photon track reconstructed in the GLAST LAT. 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 of the sources. The informations used by this algorithm are the hit in the LAT tracker generated by the Digitization package. The basic idea is to skip the clustering procedures, and looks only for hits in the 3 planes that have started the trigger. Only the hits aligned in a straight line, within a certain tolerance, are considered. From the total sample, the triplets sharing points are merged, in order to reduce the number of tracks. From these merged triplets, the direction of the initial photon is derived by simple vertexing stategies.

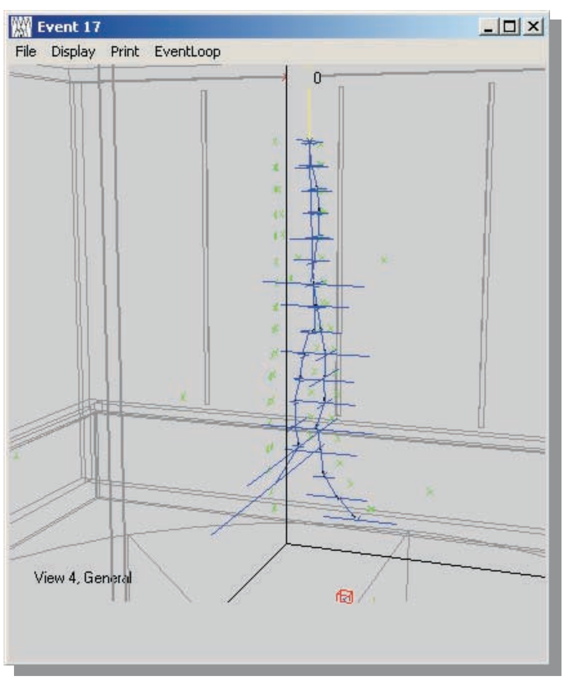

Fig. 6. 100 MeV gamma tracked in the LAT

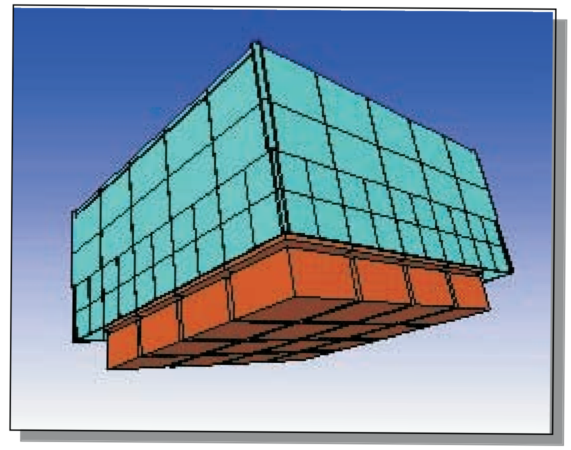

Fig. 7. GLAST LAT visualized with VRML

VI. GEOMETRY DATABASE ACCESS LIBRARY

The geometrical information of the detector and its materials are stored in an unique repository written using the XML language. A series of classes have been implemented to retrieve the geometry hierarchy from these files and represent it in memory; throught the use of abstract classes and design patterns, it has been possible to design various visitors of the geometry hierarchy to accomplish different tasks like several graphical outputs, the Montecarlo geometry construction and reconstruction-digitization tasks. Figure 7 shows the external appearance of GLAST visualized with a VRML browser.

VII. 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 8 shows a recent FRED-based event display of GLAST.

VIII. A FAST SIMULATOR FOR THE SKY MAP OBSERVED BY GLAST

A program for the simulation of the sky map, in the gamma-ray energy range, observed by the GLAST experiment has been implemented. The program generates a list of photons and images of the galactic and extragalactic background and of the sources, in a selected energy range and in a given region of the sky.

The simulation program is organized as follows: the galactic background map can be generated either using the Galprop

0-7803-8257-9/04/$20.00 © 2004 IEEE.

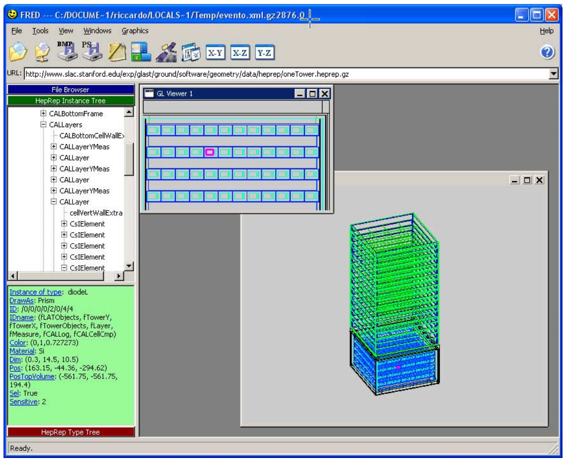

Fig. 8. GLAST LAT event display based on FRED

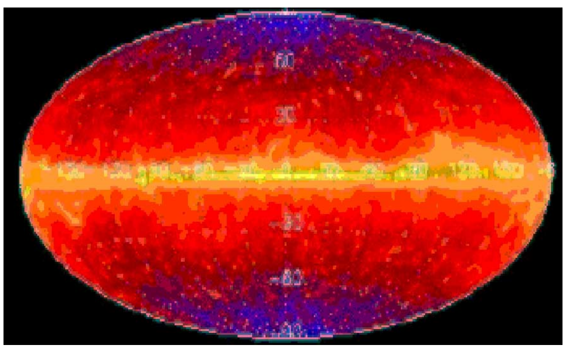

Fig. 9. Sky map seen by GLAST obtained with the fast simulator

program[17] or the model of diffuse galactic background[18] obtained using observations from EGRET. The extragalactic contribution is given by a constant value in a fixed energy range. The gamma emission of the sources is parameterised using a standard power law and sources from the Third Egret Catalog[19] and faint sources generated following the Stecker and Salamon model[20] are considered. All the contributions (background and sources) are integrated in the chosen energy range (Emin, Emax), in a fixed region of latitude and longitude (b, l) and convoluted with the instrumental point spread function, effective area and energy resolution of the LAT instrument. Finally the total intensity is multiplied by the exposure time calculated from the simulation of the orbit.

The result of the simulation yields images (FITS format) and two FITS tables containing the information required for the analysis. In Figure 9 the maps with the contribution from all sources plus background are shown for the full sky after 55 days (one precession period) and in the energy range 0.1 30 GeV.

IX. Conclusions

The *Gleam* simulation program has been developed in the last few years and now it's ready for simulating the full GLAST satellite and is being used for deriving the final instrumental parameters (useful for example for specific analysis such as that reported in[21]) and for generating a full set of events for the developing of scientific analysis software.

References

[1] http://glast.gsfc.nasa.gov/
[2] GLAST Science Brochure (March 2001), http://glast.gsfc.nasa.gov/science/resources/
[3] P.Boinee et al. (2003), "Gleam: the GLAST Large Area Telescope Simulation Framework", in "Science with the New Generation of High Energy Gamma-ray Experiments. Between Astrophysics and Astroparticle Physics", eds. S. Ciprini, A. De Angelis, P. Lubrano, O. Mansutti, Forum Editrice Universitaria, Udine, arXiv:astro-ph/0308120
[4] Gaudi project, http://proj-gaudi.web.cern.ch/proj-gaudi/
[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 toolkit home page: 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, "The FRED Event Display: an Extensible HepRep Client for GLAST," arXiv:cs.gr/0306031. Talk given at 2003 Conference for Computing in High-Energy and Nuclear Physics (CHEP 03), La Jolla, California, 24-28 Mar 2003. Published in eConf C0303241:MOLT010,2003
[14] http://www.fisica.uniud.it/~riccardo/research/fred
[15] J. Perl, R. Giannitrapani and M. Frailis, "The Use of HepRep in GLAST," arXiv:cs.gr/0306059. SLAC-PUB-9908, CHEP-2003-THLT009, Jun 2003. 9pp. Presented at 2003 Conference for Computing in High-Energy and Nuclear Physics (CHEP 03), La Jolla, California, 24-28 Mar 2003.
[16] M. Dönszelmann, A. Ballaminut, C. Colonello, E. van Herwijnen, D. Köper, J. Korhonen, M. Litmaath, J. Perl, A. Theodorou, D. Whiteson, E. Wolff "WIRED - World Wide Web Interactive Remote Event Display", Prepared for International Conference on Computing in High-Energy Physics and Nuclear Physics (CHEP 2000), Padova, Italy, 7-11 Feb 2000
[17] A.W. Strong and I.V. Moskalenko (2001), "New Developments in the GALPROP CR Propagation Model", arXiv:astro-ph/0106504
[18] S.D. Hunter et al. (1997), "EGRET Observations of the Diffuse Gamma Ray Emission from the Galactic Plane", ApJ **481**, 205
[19] R.C. Hartman et al. (1999), "The Third Egret Catalog of High Energy Gamma Ray Sources", ApJ Suppl. **123**, 79
[20] F.W. Stecker and M.H. Salamon (1996), "The Gamma Ray Background from Blazars: a New Look", ApJ **464**, 600
[21] A. Cesarini *et al.* (2003), arXiv:astro-ph/0305075, submitted to Astroparticle Physics