GLAST Large Area Telescope:
Overview of GLAST Offline Software

Richard Dubois
Stanford Linear Accelerator Center
richard@slac.stanford.edu

Representing the GLAST Software Group

http://www-glast.slac.stanford.edu/software
Outline

• The GLAST mission and instrument
• Introduction to GLAST simulation and reconstruction
• Beg Borrow and Steal  (® Bob Jacobsen)
• Use of Gaudi
  – Tools
  – G4 interface
  – Root I/O
  – Calibration Infrastructure
• Track Reconstruction in a (massive) pair conversion telescope
• Links to other GLAST talks at CHEP03
• Summary
GLAST measures the direction, energy and arrival time of celestial gamma rays

- LAT measures gamma-rays in the energy range \( \sim 20 \text{ MeV} - >300 \text{ GeV} \)
  - There is no telescope now covering this range!!

- GBM provides correlative observations of transient events in the energy range \( \sim 20 \text{ keV} - 20 \text{ MeV} \)

Launch: September 2006
Florida

Orbit: 550 km,
28.5\(^\circ\) inclination

Lifetime: 5 years
(minimum)
GLAST Instrument: Large Area Telescope (LAT)

- Array of 16 identical “Tower” Modules, each with a tracker (Si strips) and a calorimeter (CsI with PIN diode readout) and DAQ module.

- Surrounded by finely segmented ACD (plastic scintillator with PMT readout).
Sim/Recon Toolset – Beg, Borrow and Steal

- unique to GLAST

Root, IDL – analysis

TkrRecon, CalRecon, AcdRecon, Astro sources

GEANT4 – simulation package

xml – geometry, parameters

Root – object I/O

Gaudi – code framework

doxygen – doc

VC++ – Windows IDE
gnu tools - Linux

vcmt – Windows, Linux gui

CMT – package version management

cvsweb – www view of repo

cvs – file version management

ssh – secure cvs access

applications

utilities
Example of Using Gaudi Tools

```c
IEnergyCorr* m_lastLayerTool;
sc = toolSvc()->retrieveTool(m_lastLayerToolName, m_lastLayerTool);

m_lastLayerTool->setTrackSlope(slope);
m_lastLayerTool->doEnergyCorr((*it)->getEnergySum(),(*it));
```

Retrieve tool by name via base class

Refer to base class functions.
Does not know which concrete tool it is.

Tools id’ed by name in ascii config file (“jobOptions”)

Concrete classes that customize behaviour

**jobOptions**

**Parameters:**

- **CalClustersAlg.callNumber**
  this parameter is used to distinguish multiple calls to **CalClustersAlg**
  (for example, before and after TkrRecon). The default value is 0.

- **CalClustersAlg.clusterToolName**
  name of tool performing clustering.
  Default is **SingleClusterTool**

- **CalClustersAlg.lastLayerToolName**
  name of tool performing last layer energy correction

- **CalClustersAlg.profileToolName**
  name of tool performing profile fitting energy correction
Gaudi Interface to Geant4

http://www-glast.slac.stanford.edu/software/core/documentation/reviews/G4Generator/g4greview.pdf
Instrument Simulation and Reconstruction

3 GeV gamma interaction

Source Fluxes

Particle Transport

“Raw” Data

Instrument data

Recon

3 GeV gamma recon

Background Rejection - Particle ID

Full geometry in xml with C++ interface
G4 discovers instrument from the xml

CAL Detail
Data flow in the Gaudi framework

- Simulation Algorithms
- Source Generators
- Digitization Algorithms
- Reconstruction Algorithms
- Ntuple Service
- Ntuple

- Raw data
- Level 0
- Pseudo Persistency Algorithms
- Level 1

- Ready for astronomy

ACD, TKR, CAL

MC
Real Data

G4

R.Dubois
RootIo – No TBlobs for Us

Writing
- mc.root
- digi.root
- recon.root

Reading
- mc.root
- digi.root
- recon.root

GLAST Event Data Store
- mcRootWriterAlg
- digiRootWriterAlg
- reconRootWriterAlg
- mcRootReaderAlg
- digiRootReaderAlg
- reconRootReaderAlg

RootIo
Problems and a Possible Solution

• Use of algorithms is inconsistent with the spirit of Gaudi’s Persistency Service.
• Does not provide fine control over what is read/written – it’s all or nothing as currently implemented.
• Monolithic algorithms are more difficult to maintain versus light weight converters.

• There is a “real” ROOT service under development
    -ROOT I/O
    -ROOT interactive session by demand
    -ROOT share library dynamic loading by demand
    -ROOT control over the Gaudi algorithms

• We hope to use this code directly, or modify it for our needs.
Calibration Infrastructure Diagram

See talk by Joanne Bogart

Calibrator

I & T Client

Gaudi Client

calibUtil interface

Write/register
Search
Read

MySQL rdbms

Metadata (persistent)

Data (persistent)

bad strips
(XML)

CAL calibs
(ROOT)

ROOT, XML
services

R.Dubois
**Tracker/Converter Issues**

At low energy, measurements at first two layers completely dominate due to multiple scattering--

Low energy PSF completely dominated by multiple scattering effects:

\[ \theta_0 \sim 2.9 \text{ mrad} / E[\text{GeV}] \]

(scales as \((x_0)^{1/2}\))

High energy PSF set by hit resolution/Plane spacing:

\[ \theta_D \sim 1.8 \text{ mrad} \]

At 100 MeV, opening angle \(\sim 20\) mrad

At higher energies, more planes contribute information:

<table>
<thead>
<tr>
<th>Energy</th>
<th># significant planes</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 MeV</td>
<td>2</td>
</tr>
<tr>
<td>1 GeV</td>
<td>(~5)</td>
</tr>
<tr>
<td>10 GeV</td>
<td>&gt;10</td>
</tr>
</tbody>
</table>

All detectors have some dead area: if isolated, can trim converter to cover only active area; if distributed, conversions above or near dead region contribute tails to PSF unless detailed and efficient algorithms can ID and remove such events.
Multiple Scattering in Converter Layers

- 100 MeV gammas
  - Mean angle: ~17 mr
  - Separation at next layer: ~550 µm
  - Strip pitch 228 µm
  - Barely resolvable into separate strip hits @100 MeV!

- MS blows up the opening angle significantly!
  - Mean angle: ~140 mr
  - Separation at next layer: ~4.5 mm
  - Easily resolvable

- Note design:
  - Blue is “front” 12 3% X0 layers
  - Green is “back” 4 25% X0 layers
    - Last 2 have no radiator
  - To optimize interaction rate vs resolution

Multiple scattering critical to tracking at low E!

Use Kalman filter to account for large MS contributions
Tracking Reconstruction Example

100 MeV Gamma
GLAST Talks at CHEP03

- Simulation/Reconstruction Overview – R.Dubois
- System Tests and Build Environment – K.Young
- Calibration Infrastructure – J.Bogart
- GUIs on CMT – T.Burnett
- HepRep for GLAST – J.Perl
- FRED Event Display – R.Giannitrapani
Summary

• GLAST sim/recon has same problems as “the big boys”

• Adopted HEP standards
  – GEANT4, Gaudi, Root, CLHEP, CMT
  – Flexible geometry in xml to describe beam tests & flight unit without code changes

• Added user interfaces on top of CMT

• Pair converter recon is unique to GLAST