GLAST tracking software

W. Atwood, J. A. Hernando, R. P. Johnson, H. Sadrozinski

University of California, Santa Cruz
Tracking simulations
Software workshop, September 7th, 2000

Introduction:
The tracker software in GLAST
  the detector
  the physics
  the evolution of the GLAST code
  the general requirements

Simulation Part:
  Status
  some ideas for the future

Reconstruction:
  Introduction:
    code evolution and versions
    physics of the reconstruction
    status
    code structure
    performance
    future planes
    data structure
    package configuration
    conclusions
Tracking simulations
GLAST tracker/converter

GLAST tracker:
- A modular design - 4x4 towers
- each tower has 18 active planes (trays)
- microstrips silicon detectors (200 micros) with digital readout
  per silicon plane: ToT and a list of strips
- Each tray separated 3.2 cm
  XY projections - separated by 0.24 cm
  Pb converters (2.5%-25%) just above the Si.
  1.5% XO of support material per tray

Elements of the tray:

U. California, Santa Cruz
GLAST gamma converter

* GLAST is a tracker/converter
  “large” amount of material 1.5 Xo (TBR)
* 20MeV-200 GeV Range of energy
  the MS effect varies from dominant to irrelevant
* Tracking electrons.
  In addition with MS we have brems
meaning of “chiSq”, “energy”, “fit parameters”.
* Gamma conversion
  Extra material
  Energy of the tracks unknown: how to get the direction of the gamma.
* Should deal with cosmic and electrons
* Should provide a trigger (3 in a row, L3T)

Design concepts:
small converter material outside the Pb physics compromise:
to gain in PSF - small converter material
to gain in Aeff - large converter material
small pitch - vertex determination
  better PSF

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
GLAST software evolution:

* until 1998 B. Atwood and T. Burnett
  authors of GLASTsim.
  initial studies.
  B. Atwood responsible of the reconstruction
  responsible of the detector performance
  He was one of the persons who most had
  contributed to GLAST.

  Small group of people
  AO response
  J.A Hernando: responsible of trk recon
  detector response.

* 2000 - software group (R. Dubois)
  Arquitecture decitions:
  CMT - package reorganization
  GAUDI - data/Algorithms
  persistency: first attempts ROOT
  GEANT4 migration
  test beam:
  ROOT (root trees): SLAC, GSFC
  tb_sim: (IRF2ROOT) a version of
  GLASTsim only for the simulation (UW)
  tb_recon: (UCSC) a reconstruction program
  centella framework
  ROOT input/ouput (trees).
  J.A. Hernando (tkr recon).

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Tracking simulations
Tracker software: general requirements

Simulation:
* It should accuracy represent the tracker detector
  It should contain the relevant passive/active materials
* It should produce the detector response of the pass of charged particles
* It should provide the detector response.

Simulation - Interaction with MC:
* It should provide the MonteCarlo information of how and who generated the detector response

Simulation - Interaction with trigger:
* It should help the definition of the different triggers

Reconstruction:
* It should reconstruct the tracks produced in the event
  It should reconstruct the gamma
* It should provide the tracks reconstruction information
  It should reconstruct the physical hits

Reconstruction - Interaction with CAL/ADC
* It should provide a combined determination of energy

U. California, Santa Cruz
Tracking simulations

Simulation: status

**Status:**

* It should accurately represent the tracker detector
  
  GISMO: W. Atwood, T. Burnett
  
  parameters defined in the xml file
* It should produce the detector response of the pass of charged particles
  
  average value of occupancy and threshold
* It should provide the detector response.
  
  GLASTsim internal “persistency”: IRF files

**Known Problems**

change of the tracker detectors:

unique size of the detector

Pb constructor by reference to SI

not definition of some tracker elements (face sheets, electronics).

IRF decodification Not real persistency output of the simulation.

No valid checks of the performance of the pattern recognition, reconstruction by lack of MC information.

**Simulation - Interaction with MC:**

*It should provide the MonteCarlo information of how and who generated the detector response

  only noise or real hit

**Simulation - Interaction with trigger:**

* It should help the definition of the different triggers

  U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Tracking simulations

Simulation: conclusion

Tracker simulation:
* the tracker simulation is an acceptable level but should be revisit
* a clear and well define list of geometry parameters should be defined in an input file
* a “calibration” input for the electronic parameters should be revisit.
* the MC information associated with the tracker should be expanded.
* a persistency output should be added

In the new architecture:
* the transient classes to contain the tracker response should be defined
  * pure tracker information.
  * association with the MC
* the simulation: geometry and passage of the particle will migrate to GEANT4.
* A persistency data output should be defined.

U. California, Santa Cruz

GLAST software workshop, SLAC, Sept 2000
**Tracking simulations**

**Simulation: some ideas**

**Persistency:**
* the parameters in the xml should be revisit: coordination with R. Johnson.
  * they should correspond 1-1 with the mechanical design
  * they should contain the flexibility of defining the different elements of the tray
    * dices/ladders/planes/tray
    * layers in tray/electronics
  * should we include threshold/occupancy by chip?

*The simulation should provide a useful persistency data: (i.e. ROOT tree)*
  * do we need IRF as intermediate step?
  * TB: IRF2ROOT program

**Transient Data:**
* Definition of a container class of the tracker response:
  * TB: SiLayerList
  * SiLayers info: ID, ToT, list of strips
* Definition of the tracker response-MC connection Data.

U. California, Santa Cruz
Tracking simulations

Reconstruction: evolution and versions

Tracker reconstruction evolution:

* until 1998 B. Atwood and T. Burnett

  LSQ fit
  pattern recognition based in a best track
  definition of the main physics algorithms

  Tracker reconstruction based on:
    Kalman Filter
    Pair Fit (2D-3D) Pattern recognition:
    based in a gamma vertex
    AO response

* 2000 software group (R. Dubois).
  Tb_recon
  centella framework & ROOT tree
  definition of transient classes
  first reorganization in algorithms
  converters to ROOT tree

New structure (work in progress):
  definition of data and algorithms
  reorganization in packages

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Tracking simulations

Reconstruction: summary status

Tracker reconstruction status:
* It is in a good shape to produce results (gamma and tracks reconstruction) for:
  
  AO
  
  TB
  
  recent GLAST design studies GTOCC
* The physics algorithms works: they should be preserved, better understood and optimized.

  Kalman Filter
  Pair Fit
  Energy determination
  Topological selection “pair fit”
  Selection criteria (quality && veto)
  Gamma construction
* Studies with MC needed:
  
  track purity
  
  performance of the pattern (selection of best track)

Status with respect the new structure
* the present reconstruction is unfortunately OO
  
  the tracks are pattern objects and tracking objects
  
  the tracks contain the construction/fitting algorithms
* There is no a clear user interface. How the user can retrieve tracker recon data?
  
  Not needed: the data was passed to the an ASCII ntuple
  
  GFdata is a user interface
  
  GFsegment it was an intermediate level
  
  Pattern/Kalman

The new structure
* need to preserve/improve the algorithms
* separate classes into: data algorithms conditions
* separate classes into packages
* define interfaces between packages and users
* documentation with doxygen
Tracking simulations
Reconstruction: Packages

Packages:

TkrRecon:
- It contains the transient data
- It defines the interface with the user

TkrPattern
- It contains algorithms to select and construct TkrPattern data
- A TkrPattern data is a collection of SiClusters

TkrKalman
- It contains the algorithms to fit (filter/smooth) a Kalman Track

TkrNavigator
- It contains the algorithms to navigate a TkrParticle though the detector.

Requirements:
- they adopt data/algorithm/condition separations
- they should minimize the relation with other packages
- the relations with other packages are via intermediate levels

U. California, Santa Cruz
Tracking simulations
Reconstruction: data, algorithms, conditions

GAUDI/Centella Philosophy: divide classes into data, algorithms (+ conditions)

Data:
- I.e. containers
- divided:
  - elements
  - server/List
- Base class:
  - clear()

Algorithm:
- I.e. constructors
  - Base class:
    - run()
  - Set methods
    - setData()
    - setCondition()
- Complex algorithms
  - multiAlgorithm
  - conditional Algorithms

Condition:
- bool apply()

Conditions:
- Filters returns true/false
- Base class:
  - bool apply()
- Set methods
  - setData()
  - setCondition()

Complex Conditions
- multi Conditions (AND, OR)

U. California, Santa Cruz

GLAST software workshop, SLAC, Sept 2000
Tracking simulations
Reconstruction: interface

Interface: **TkrParticle** (from GFdata)

Most of the Tkr classes should be converter into a TkrParticle: SiTrack, SiGamma, KalTrack, etc..
The connection between the different Tkr recon packages should be done using a TkrParticle
Do we need a TkrParticleComposite?

### Package Parameters:

Every package has a class with static parameters should be accessible by the user (xml file)

U. California, Santa Cruz  

GLAST software workshop,  
SLAC, Sept 2000
Tracking simulations
Reconstruction: transient data

Transient Data:

- TkrRecon
- SiLayerList: SiLayer: ToT, stripsList
- SiClusterList: SiCluster SiID, position
- SiTrackList: SiTrack: locator, quality SiHitList
- SiGamma: locator, trackList
- SiHit: cluster in track quality
- Pcandidates: list of PObjects
- P3DTrack: list of PHits
- P3DVertex: PVertex list of PTracks
- PHit: a siCluster
- KalTrack: list of KalPlanes
- KalPlane: KalHits in plane
- KalHit: parameters, Cov
- TkrParticle: Locator SiID Quality

U. California, Santa Cruz

GLAST software workshop, SLAC, Sept 2000
Tracking simulations
Reconstruction: algorithms

Algorithms

TkrRecon

- makeClusters
  - create siclusters
  - fill siclusterList

- makeSiTracks
  - create siTracks
  - fill siTrackList

- makeGamma
  - create siGamma
  - fill siclusterList

TkrPattern

- selector
  - create PObjects

- PTrackFinder
  - finde of a track
  - PTrackStep
    - step of the track

- PVertexFinder
  - finde of a vertex
  - PTrackStep
    - step of the vertex

TkrKalman

- KalFilter
  - fit a KalTrack

TkrNavigator

- TkrNavigator
  - extrapolate a TkrParticle

Base class:
- PFinder
- Pstep

Implementation

U. California, Santa Cruz

GLAST software workshop, SLAC, Sept 2000
Tracking simulations
Reconstruction: algorithms

Algorithms in the Pattern Recognition

While Algorithm:

execute() method

runs an algorithm while a condition is true

We can create Composites of WhileAlgorithms

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Tracking simulations
Reconstruction: conditions

**Algorithms**

- TkrRecon
- TkrPattern
- TkrKalman
- TkrNavigator

- PVertexAcceptor
  accept a PVertex

- PTrackAcceptor
  accept a PTrack

- PSegmentAcceptor
  accept a segment PTrack

- PHitAcceptor
  accept a PHit

- KalTrackAcceptor
  accept a valid KalTrack

---

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Tracking simulations
Reconstruction: summary of plans

Work in progress:

**user requirement document:**
- what the user can do with the Tkr recon?
- How the user retrieve data from Tkr recon?

**Reorganization in packages:**
- Kalman Filter - Pattern Recognition
- TkrRecon - TkrNavigator
- 60% work done
- Waiting to commit to CVS.

**Coordination with SLAC**

**Definition of data and algorithms:**
- **tb_recon:**
  - first attempt data reorganization: SiClusters

**Future Planes:**
- user requirements document
- tracker reconstruction documentation
- adaptation of the code to the new structure
- tracker recon group: UCSC, USC, SLAC
- goal: next release (jan 01).

U. California, Santa Cruz

GLAST software workshop,
SLAC, Sept 2000
Conclusions:

* we need to understand the present reconstruction in terms of the general GLAST software evolution.

* the tracker reconstruction works in an acceptable level and has produced good results for:
  
  AO, tb_recon, and GTOCC.

* the tracker reconstruction contains a collection of correct algorithms for the GLAST tracking problem
  
  Kalman Filter      Pair Fit (“V” pattern recognition)

* the tracker reconstruction contains a minor collection of physics algorithms that should be understood, and explored.
  
  Best track, addition of tracks, energy determination

* the tracker reconstruction should be reorganize to fit into the new structure. Work in progress.