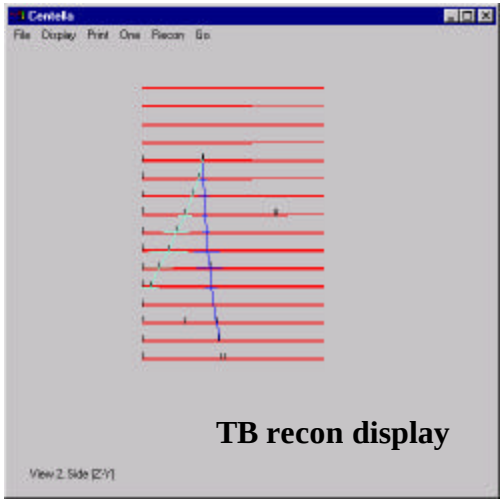
Tb Recon - Centella

Software workshop, sep 7th, 2000

Tb_recon *Centella*

A Reconstruction Program for the 99/00 GLAST Test Beam



TB recon display

TB RECON

- TB Recon - Structure
- TB Recon - How To
- TB Recon - conclutions

Jose A. Hernando, ***UCSC***

R. Dubois, D. Flath, P. Kunz ***SLAC***

A. Djannati, R. Terrier, ***Ecole de France***

H. Arrighi, ***GSFC***

U. California, Santa Cruz

Tb Recon - Centella

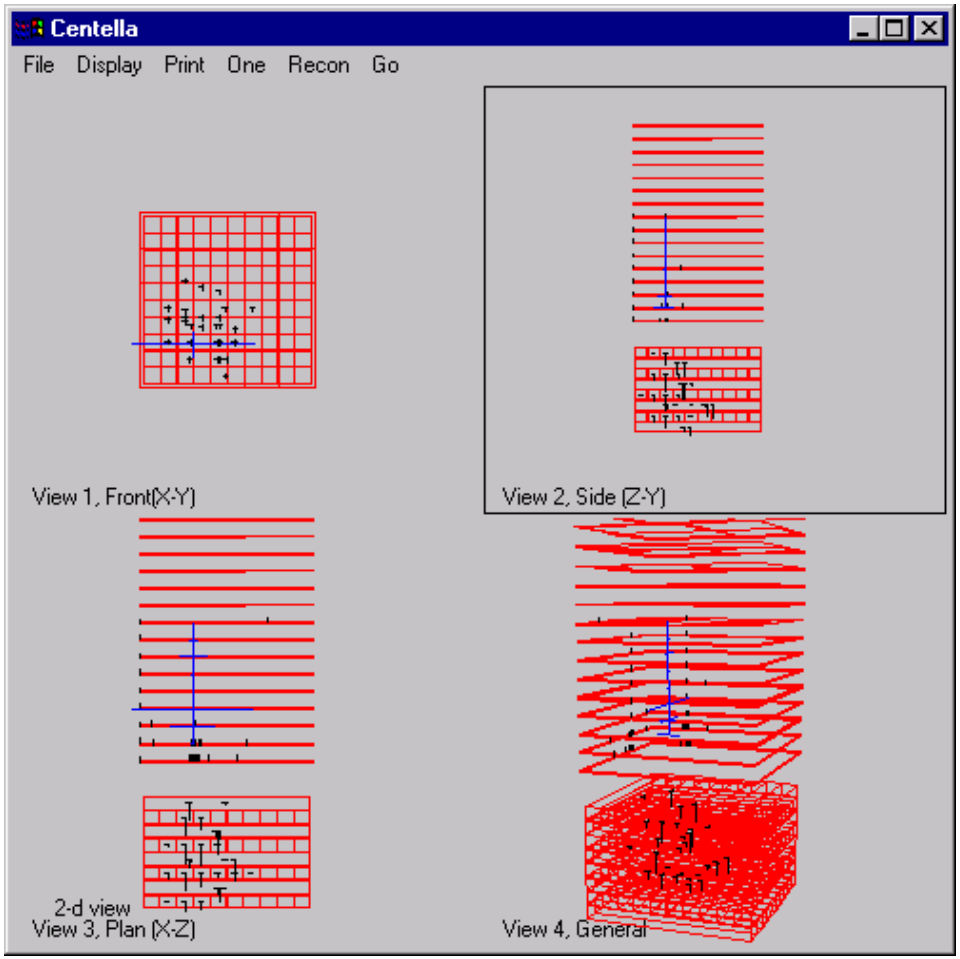
Tb_recon - Centella

- **Centella**
  - It is now a package under CVS
  - It defines a framework
    - uses GAUDI philosophy

* **Centella Follows Gaudi's philosophy**

- It Separate Data/Algorithm
- It Separate Data in persistent (ROOT)-transient.
- It provides servises: Option, message.

- **TBRecon uses centella as framework**
- **TBRecon - Purposes:**
  - Perform a full reconstruction
  - Can be use to particular detector studies
- **It inheretes from GLASTSIM**
  - Win display 3D
  - We use the GLASTsim Tracking Reconstruction



Event Display: a high energy Gamma conversion, run 300

Tb Recon - Centella

Tb_recon - structure

Application:

Centella (algorithm)

- Data Manager

  List of Data

- Process Manager

  List of Algorithm

- Selection Manager

  List of Cuts

- Option Server

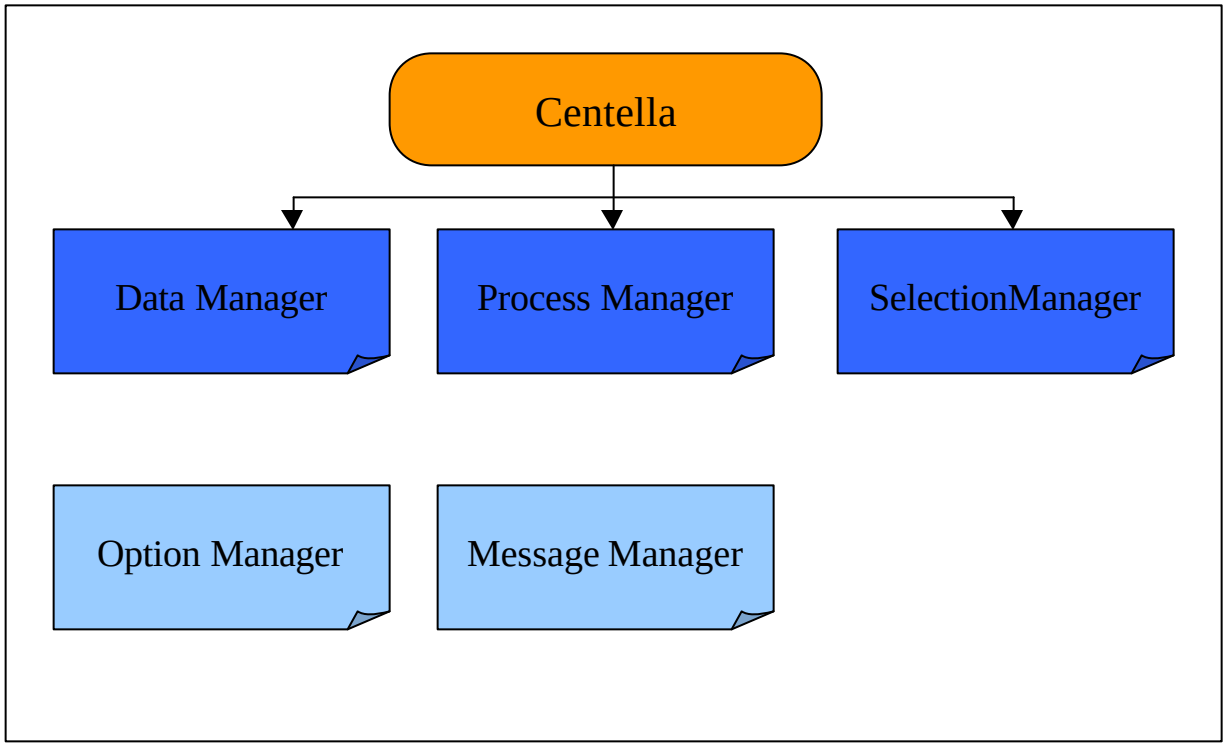
  (set option by the user)

- Message Server

  (print debug, info,etc)

- Selection Server

  (apply selection criteria)

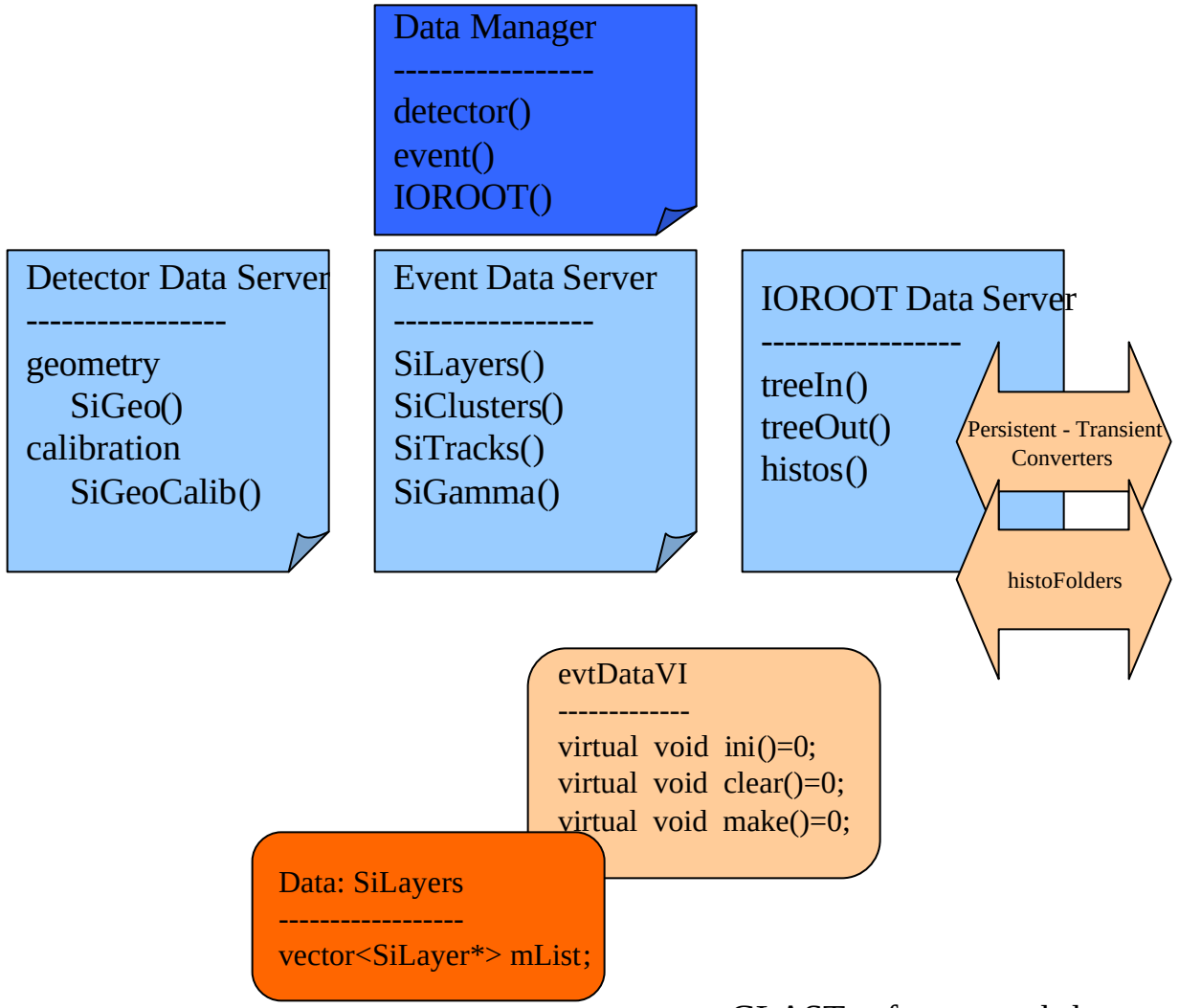


Tb Recon - Centella

Tb_recon - Data Manager

Data Manager

- **Persistent Data**
  - Root Trees (I/0)
  - histograms (0)
- **Transient Data**
  - **Run Data**
    - geometry
    - calibration
  - **Event Data**
    - generation
    - reconstruction
    - analysis
- Transient Data inherits from
  - **trsDataVI** (virtual)



Tb Recon - Centella

Reconstruction: data,algorithms, conditions

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

Data:
make();
clear();

Data:

I.e containers

devided:

elements

server/List

Base class:

clear();

Algorithm:
run();
initialize();
execute();
finalize();

Algorithms:

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)

Tb Recon - Centella

Tb_recon - Managers

The managers contain a template server serverVI *map<string,T>*

Process Manager	Data Manager	Selection Manager
addAlgoritm();	addData();	addCut();
getAlgorithm();	getData();	getCut();

How to implement an algorithm:

1) create an algorithm class that inherits from algorithmVI

2) declare the class to the processManager

(it ca use the class userAlgorithms)

How to implement transient data:

1) create a data class that inherits from trsDataVI

2) declare the class to the dataManager

(it ca use the class userDataEvent or userDataDetector)

How to implement a cut (condition):

1) create a cut class that inherits from algorithmVI

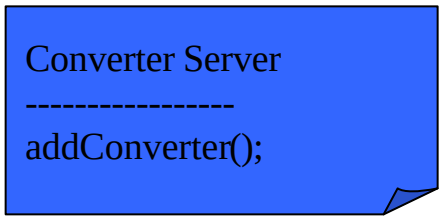
2) declare the class to the selectionManager

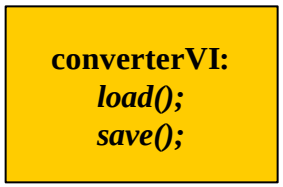
(it ca use the class userCuts)

Tb Recon - Centella

Tb_recon - Converters

The server contain a template server serverVI *map<string,T>*





How to implement a converter:

1) create a converter class that inherits from converterVI

(I.e SiLayers)

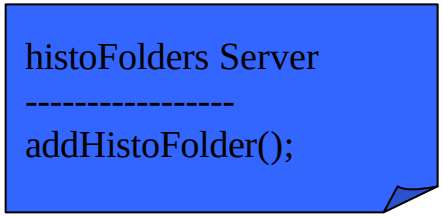
2) declare the converter to the server

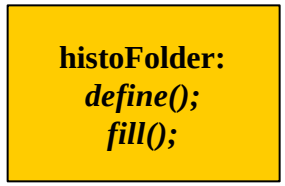
addConverter("SiLayers", new SiLayers());

3) it will automatically create two algorithms named:

loadSiLayers and saveSiLayers

available in the processManager





How to implement a folder with histograms and ntuples (ROOT)

1) create a cut class that inherits from histoFolder

I.e ToThistoFolder

2) declare the class to the histoFolder server

addHistoFolder("ToThistoFolder", new ToThistoFolder());

3) It will automatically create an algorithms named

fillToThistoFolder

available in the processManager

(the fill() method will call the define() (or book)

Tb Recon - Centella

Tb_recon - set options

Option Manager
---
addOption();
setOption();

optionVI:
setName();
defineOption();
setOption();

•It contain a template server serverVI *map<string,optionVI*>*

• setOption()

read the options from the centella.in (or the user input file) and set them into one of the objects in the server

```
// this is a comment in the input file (centella.in)

// this set an option
SiAlignment  fileName S D:/planeOffset.txt

// you can set double D, integers I, or string S
```

How to make a class member variable an option that the user can modify using centella.in

1) create an algorithm class that inherits from serviceI (or optionVI) (I.e CalCalib)

2) set a name to the class setName("CalCalib");

(it automatically gets declared to the Manager)

3) convert a member variable into an option:

use this method in the constructor of the class

defineOption("fileName",&m_fileName);

Tb Recon - Centella

Tb_recon - dynamic algorithms and cuts

The user can create dinamically algorithms and cuts, or a combination of both

Using the centella.in (or input file)

The user can also decide the application algorithm to run (default centellaAlg)

Process Manager

The user can create

AnAlgorithmTask

An eventTaks

```
processManager newAlgotihmTask S myTask
myTask addAlgorithm S openInputFile
myTaks addAlgorithm S openOuputtFile
```

```
processManager newEventTask S myEvent
myEvent  generation S loadSiStrips
myEvent reconstruction S makeSiCluster
myEvent reconstruction S makeSiTracks
myEvent analysis S fillSiHisoFolder
myEvent analysis S saveSiTracks
```

Selection Manager

The user can create

A **selectionTask**

```
selectionManager newSelectionTask S mySelection
mySelection  addCut S cutGoodGamma
mySelection  addCut S cutGoodTrackerReconstruction
```

The user can create

An **algorithmConditionalTask**

Tb Recon - Centella

Tb_recon - centella

Centella Conclusions:

Centella is a simple package (few docens of classes).

centella is a simple framework that mimic the GAUDI philosophy

separation data/algorithms/(+ cuts)

centella provide a simple way to set options from a external file

optionManager

centella does not need “persistency” structures (they use load/save algorithms)

centella is highly flexible:

centella has a large room for improvements:

there are complicated base algorithm/conditions to create

the migration from centella to GAUDI should be in principle simple.

Tb Recon - Centella

Tb_recon - centella

Tb_recon Conclusions:

tb_recon has a bad geometry detector (should be the same as the simulation, based in GEANT)

tb_recon handle poorly some aspects: GUI, exceptions.

tb_recon has a ROOT tree input

tb_recon handle the calibration files (calorimeter coeficients, silicon bad channels, aligment etc).

tb_recon produces a ROOT tree output or/and a collection of histos/ntuples

tb_recon is highly flexible

- **simple to modify algorithm (example: Ian's alignment - algorithm to find tracks)**
- **simple to add cuts**
- **simple to modify the input files (they do not need to be ROOT- simple change the converters)**

tb_recon contains the newest version of the reconstruction

- **it performs the calorimeter and/or tracker (or the algorithms user indicates in centella.in)**

tb_recon performs almost the same task that the final RECON program will do

tb_recon is almost GLAST RECON program!.