

# MarlinReco

A Marlin based  
Reconstruction Toolkit  
for the ILC

Frank Gaede  
DESY

ECFA ILC Workshop, Vienna  
August 14-17, 2005

# Outline

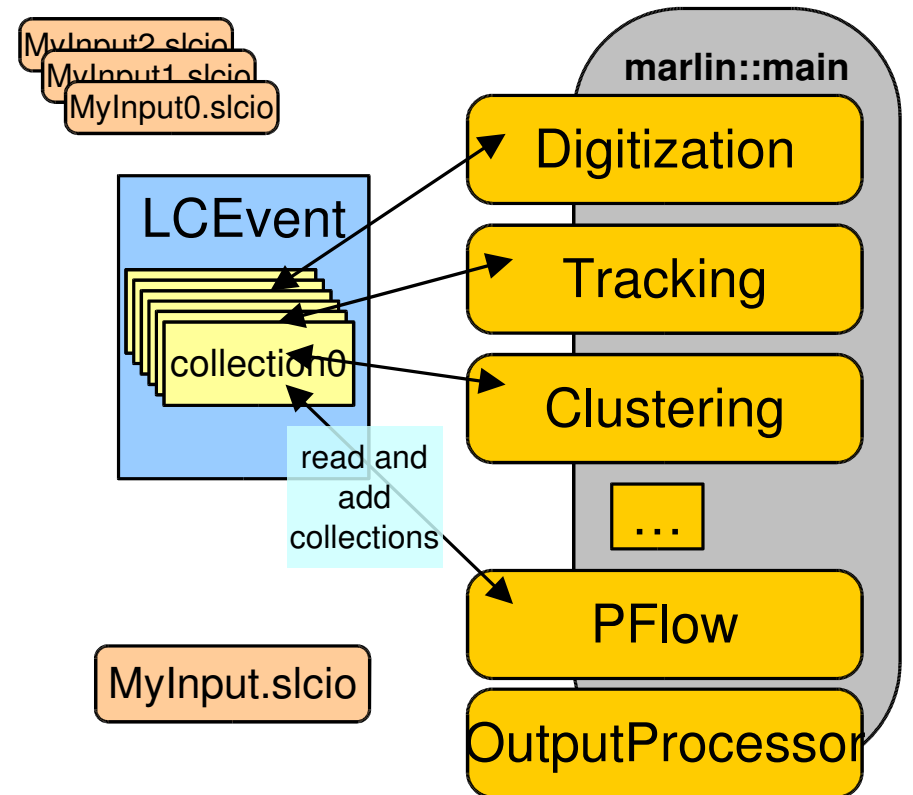
- Introduction
  - Marlin, Gear
- Overview MarlinReco
  - Reconstruction algorithms
  - PFlow
  - Utilities
- Status
- Summary/Outlook

MarlinReco developers:  
S. Aplin, F. Gaede, T. Kraemer, P. Krstonosic,  
A.Raspereza, J. Samson, O. Wendt

# Marlin

**M**odular **A**nalysis & **R**econstruction for the **L I N**ear Collider

- modular C++ **application framework** for the analysis and reconstruction of LCIO data
- uses **LCIO** as transient data model
- software modules called Processors
- provides main program !
- provides simple user steering:
  - program flow (active processors)
  - user defined variables
    - per processor and global
  - input/output files



# Gear

## GEometry API for RReconstruction

```
<gear>
  <!--
    Example XML file for GEAR describing the LDC detector
  -->
  <detectors>
    <detector id="0" name="TPCTest" geartype="TPCParameters" type="TPCParameters">
      <maxDriftLength value="2500."/ >
      <driftVelocity value="" />
      <readoutFrequency value="10" />
      <PadRowLayout2D type="FixedPadSizeDiskLayout" rMin="386.0"
        maxRow="200" padGap="0.0" />
      <parameter name="tpcRPhiResMax" type="double" > 0.16 </parameter>
      <parameter name="tpcZRes" type="double" > 1.0 </parameter>
      <parameter name="tpcPixRP" type="double" > 1.0 </parameter>
      <parameter name="tpcPixZ" type="double" > 1.4 </parameter>
      <parameter name="tpcIonPotential" type="double" > 0.00000003 </parameter>
    </detector>
    <detector name="EcalBarrel" geartype="CalorimeterParameters">
      <layout type="Barrel" symmetry="8" phi0="0.0" />
      <dimensions inner_r="1698.85" outer_r="2750.0" />
      <layer repeat="30" thickness="3.9" absorberThickness="2.5" />
      <layer repeat="10" thickness="6.7" absorberThickness="5.3" />
    </detector>
    <detector name="EcalEndcap" geartype="CalorimeterParameters">
      <layout type="Endcap" symmetry="2" phi0="0.0" />
      <dimensions inner_r="320.0" outer_r="1882.85" inner_z="2820.0" />
      <layer repeat="30" thickness="3.9" absorberThickness="2.5" />
      <layer repeat="10" thickness="6.7" absorberThickness="5.3" />
    </detector>
  </detectors>
</gear>
```

compatible with US – compact format

- well defined geometry definition for reconstruction that
  - is flexible w.r.t different detector concepts
  - has high level information needed for reconstruction
  - provides access to material properties - planned
- abstract interface (a la LCIO)
- concrete implementation based on XML files
- and Mokka-CGA – planned

# MarlinReco

- Marlin serves as a **framework** for the distributed development of reconstruction algorithms
  - provides a well defined modularity
- MarlinReco is a **toolkit** which aims at providing reconstruction algorithms for detector concept studies
  - (almost) complete set of standard reconstruction (pflow)
  - cheaters for cross checks (and replacements)
  - all processors can seamlessly be combined together with other reconstruction code or plugged into your analysis
    - e.g. together with MAGIC-clustering (see talk by C. Ainsley)

# MarlinReco packages

- **TrackDigi**

- FTDDigi
- VTXDigi
- TPCDigi

- **CaloDigi**

- LDCCaloDigi

- **Tracking**

- BrahmsTracking
- TrackCheater

- **Clustering**

- TrackwiseClustering
- ClusterCheater

- **Pflow**

- Wolf

- **Analysis**

- EventShapes
- SatoruJetFinder

most MarlinReco processors (algorithms) are geometry independent  
→ they can be applied to other detector concepts (via Gear file)

# MarlinReco Digitization

- **Tracker** (S.Aplin)
  - TPC: Gaussian smearing of position in z and r-phi
  - Silicon: exact position from simulation –
    - more meaningful clustering under development (A.Raspereza)
- **Calorimeter** (A.Raspereza)
  - calibration only - no smearing
  - optional energy cut
  - ganging (investigate different granularity)

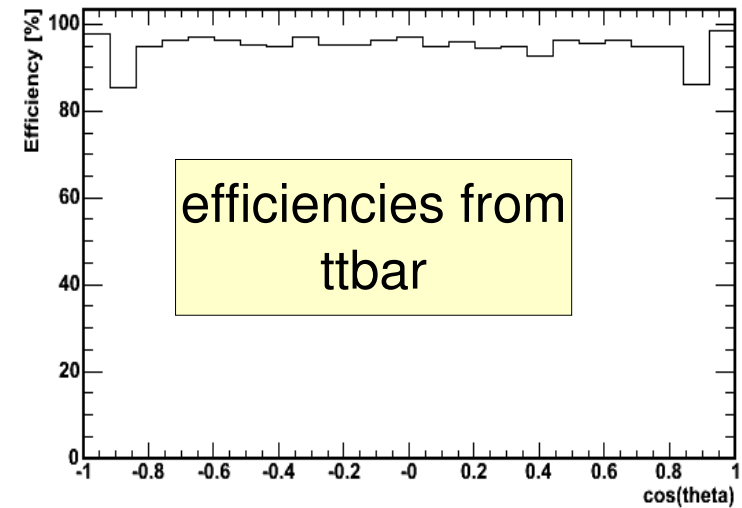
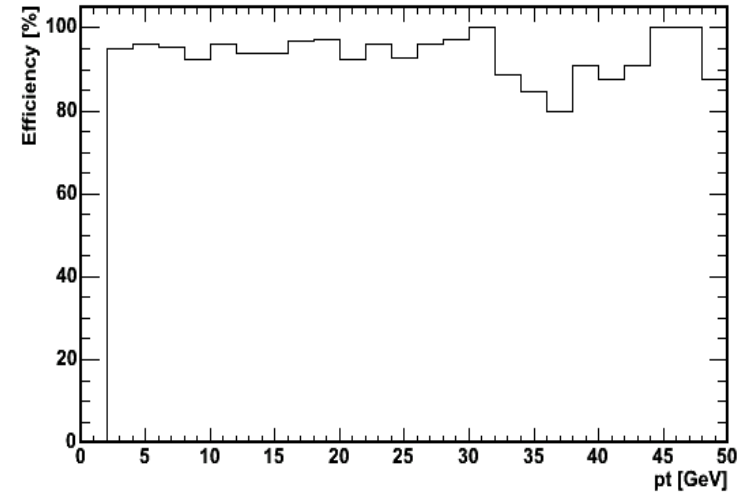
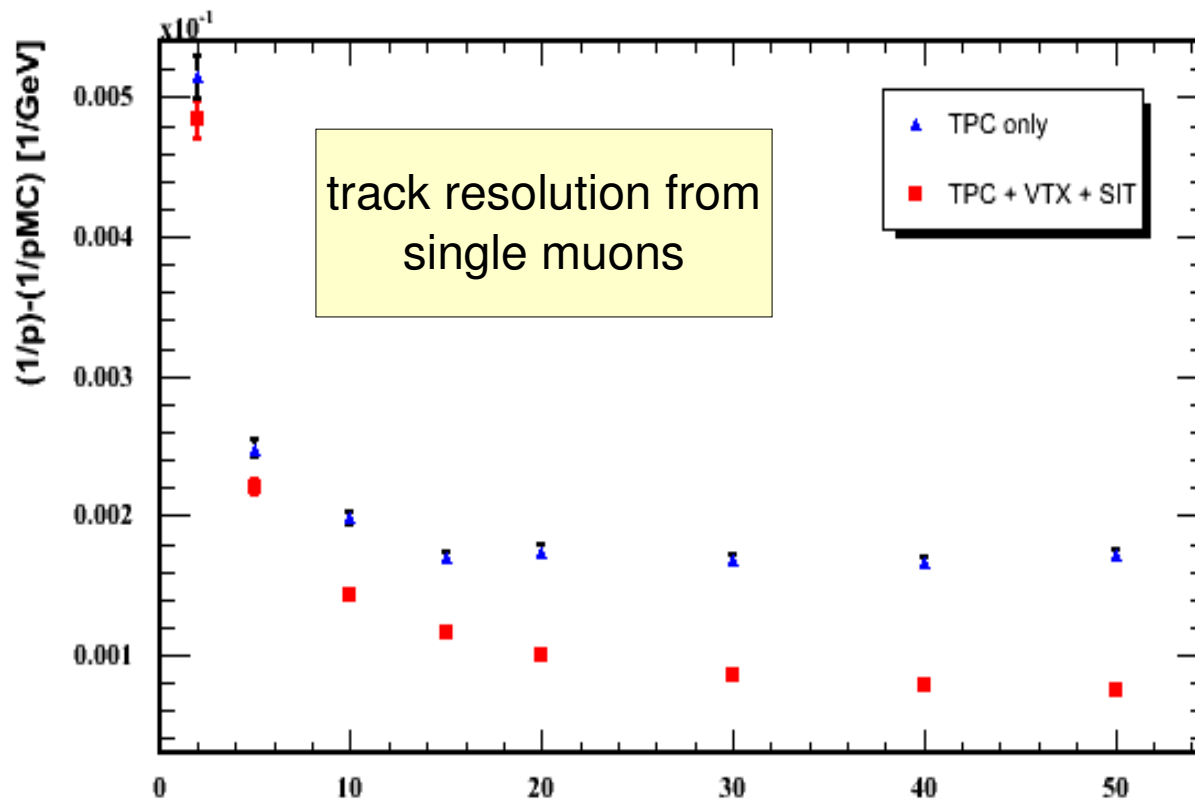
# MarlinReco Tracking

- **Central Tracks:** (S.Aplin)
  - algorithms taken from LEP (ALEPH and DELPHI)
  - f77 code from Brahms
    - track finding based on out-in search using circle fit
    - fit with Kalman Filter takes material into account
    - start with TPC tracks and then include VTX hits (next slide)
- **Track Cheater** (A.Raspereza)
  - uses Monte Carlo for track finding
  - fitted with a helix hypothesis



# Tracking including VTX hits

- vtx hits now included in track fit
- use TPC tracks as 'seed'
- standalone VTX tracking under development



S.Aplin, O.Wendt

# MarlinReco Clustering

- **Trackwise Clustering** (A.Raspereza)
  - algorithm needs spatial information only
    - -> applicable to both digital and analogue calorimeters
  - minimal dependence on detector geometry
    - -> can be used for other detector concepts
- **Cluster Cheater** (A.Raspereza)
  - uses Monte Carlo to combine hits into clusters
  - proximity criterion for 'realistic' clusters

# PFlow

- **Track-Cluster matching** (A.Raspereza)
  - extrapolate tracks into the calorimeter
    - use only outer track hits and apply helix fit
    - proximity criterion to assign cluster to track
  - charged objects (E,p) from track
  - neutral particles (E,p) from cluster
- **Particle ID** (A.Raspereza)
  - cluster shape analysis
    - fraction of energy in ECAL
    - longitudinal and transverse profile
    - test of MIP hypothesis

# Analysis

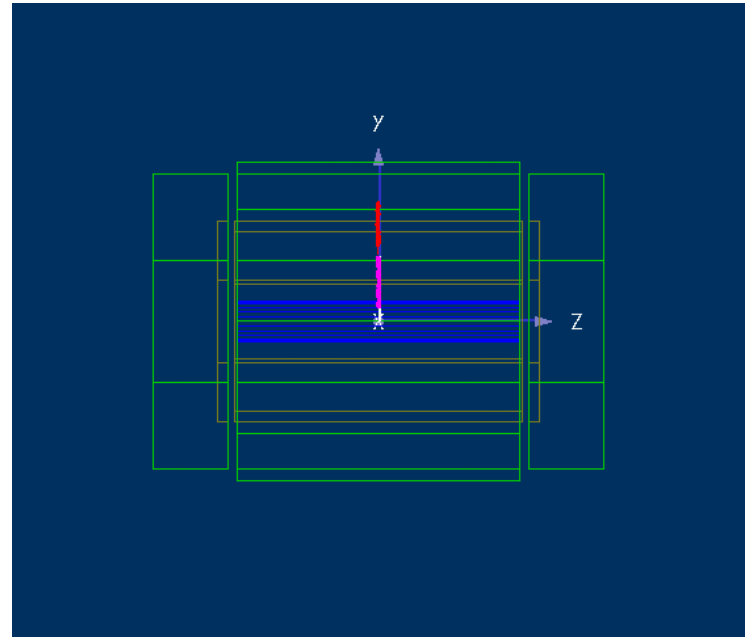
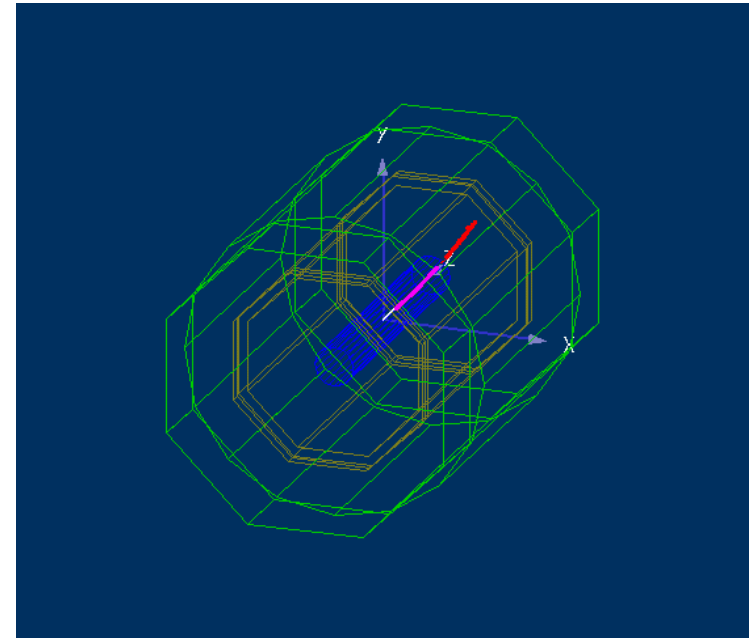
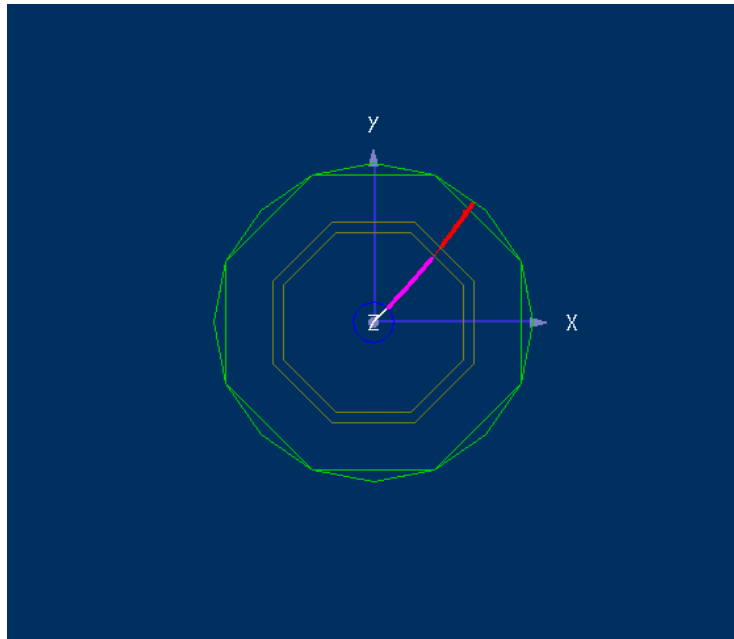
- **Event Shapes** (T. Kraemer, P. Krstonic)
  - ThrustReconstruction: Tasso & Jetnet algorithms
  - Sphere: sphericity, aplanarity, ...
- **SatoruJetFinder** (J. Samson)
  - originally developed by Satoru Yamashita for OPAL
- **BenchmarkPlots**
  - not yet -> input needed

# MarlinReco support packages

- **MarlinUtil** (O. Wendt)
  - Utility and Helper classes
    - helix fitter
    - cluster shapes
    - common code for CED
- **CED** (A. Zhelezov)
  - event display based on GLUT/ OpenGL
  - client server architecture
- **CEDViewer**
  - event display client processors

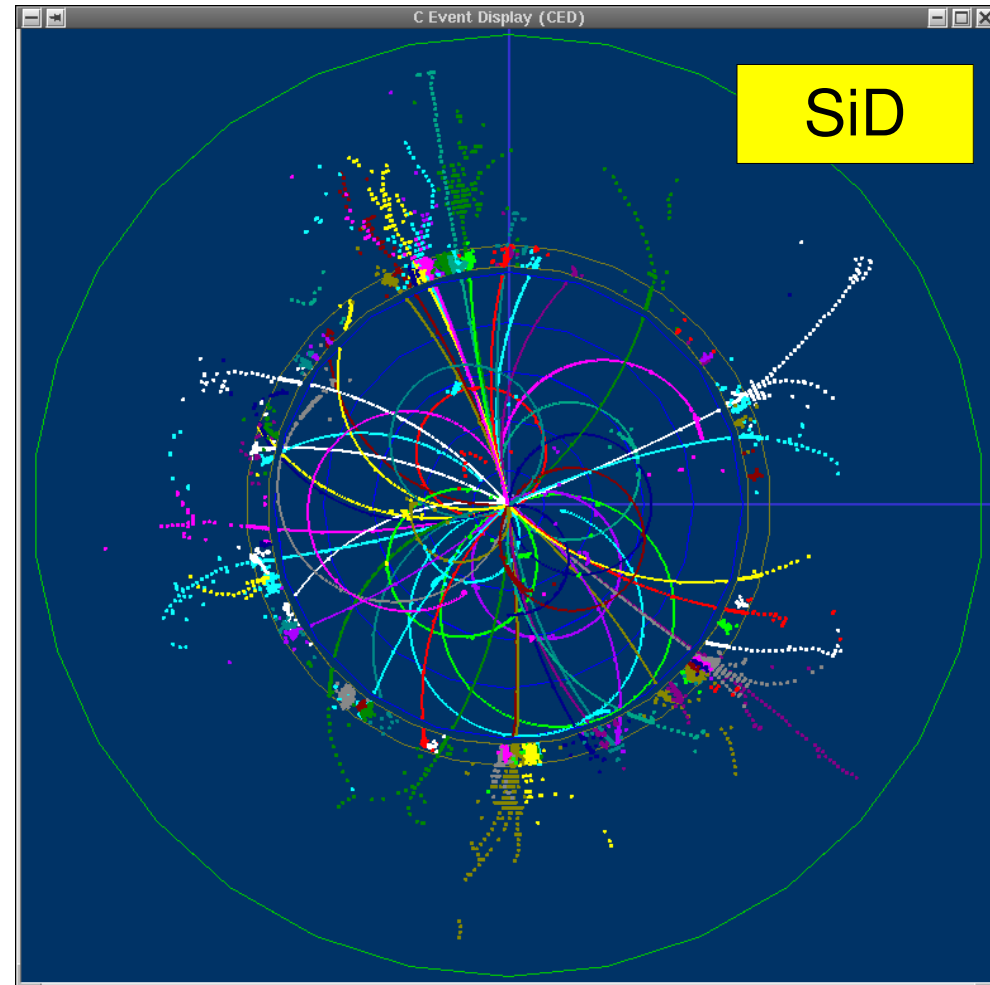
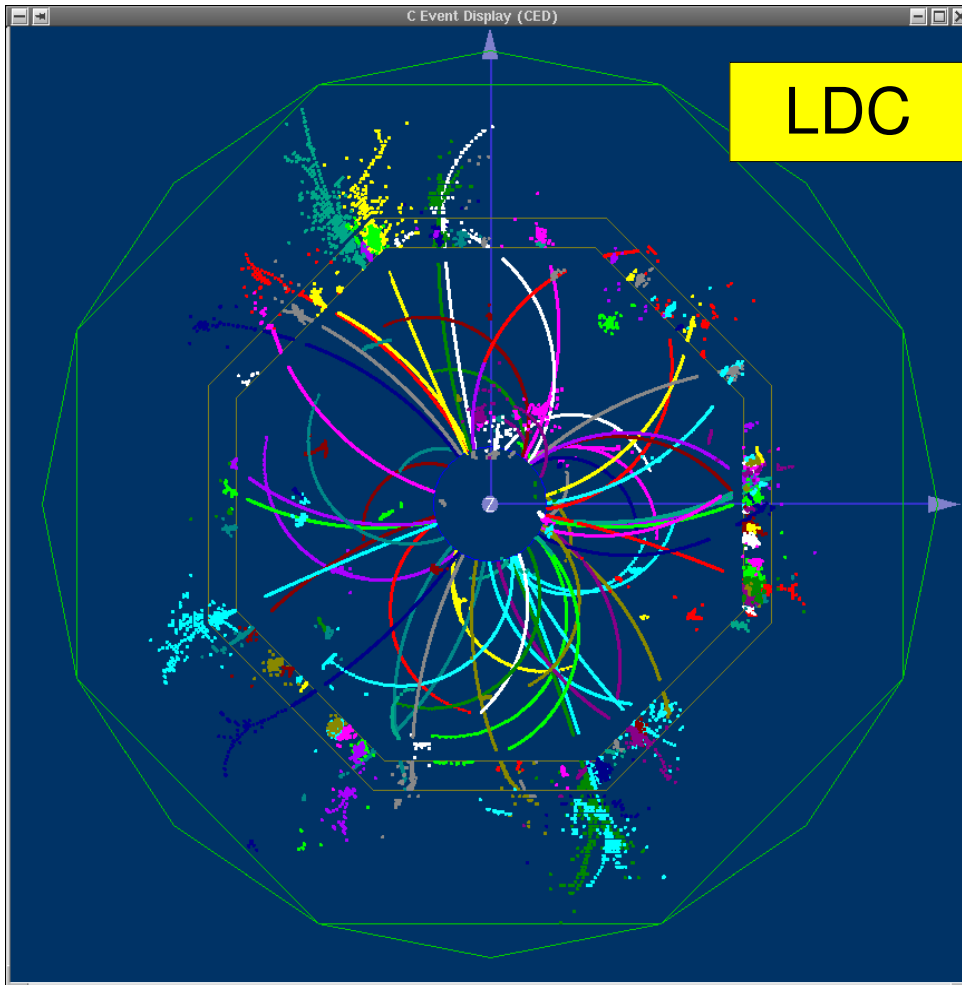
# CEDViewer

simple example taken from  
\$MarlinReco/examples/LDC  
steer\_ldc.xml  
gear\_ldc.xml



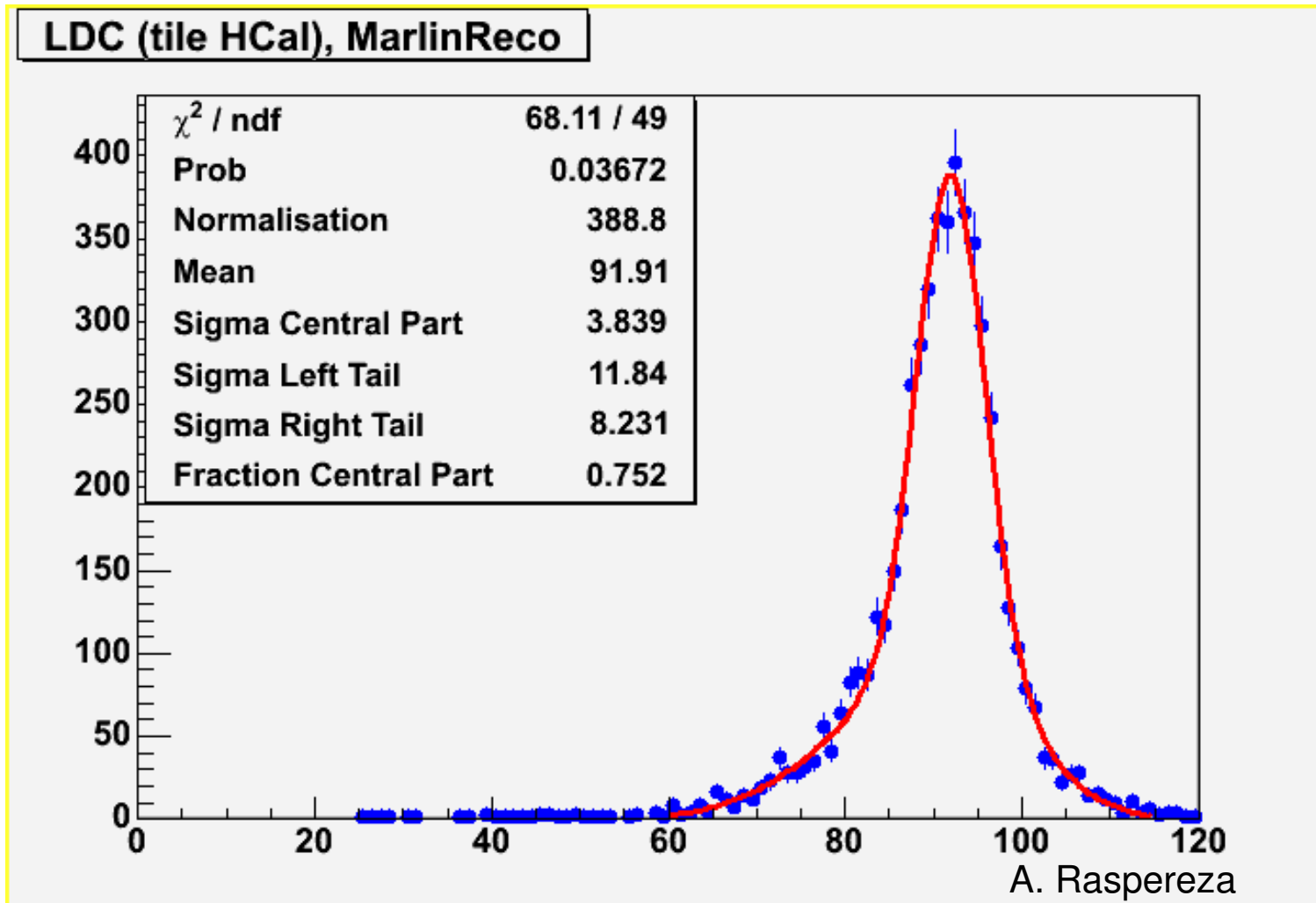
# ttbar events with MarlinReco

No cheaters, only full reconstruction



# Initial results

- $e^+e^- \rightarrow Z \rightarrow qq\bar{q}$  using full reconstruction





# Status I

- first official release of MarlinReco: v00-01
- changes wrt. snowmass DVD:
  - comprehensive manual (T.Kraemer)
  - including VTX hits for central tracking
  - detailed API documentation for every processor
- utility packages released:
  - CEDViewer v00-01
  - MarlinUtil v00-01
  - CED v00-01
- works with
  - Gear v00-02
  - Marlin v00-09-02

# Status II – To Do

- neutral vertex and kink finding algorithms
- more realistic digitization
- vertex tracking
- forward tracking
- vertexing
- particleID
- ...

your input (code) is welcome !

# MarlinReco on the web

Frank Gaede, ECFA ILC Workshop, Vienna, Nov 14-17, 2005

The screenshot shows the MarlinReco project page on the ILC Software Portal. The page features a navigation sidebar on the left with links to Home, Software packages, and various software tools. The main content area includes the project name 'MarlinReco', its category 'Reconstruction software', and a section for 'A Marlin based reconstruction software'. It highlights the 'Latest release: MarlinReco v00-01' and provides a 'Download MarlinReco for all platforms' link. A 'Project Description' section explains that MarlinReco is a software package for event reconstruction using simulated raw data. The page also includes a login form and a list of additional resources such as Documentation, Releases, and Contact address.

This screenshot displays the MarlinReco CVS repository page. It features a table listing files and their revision history. The table has columns for File, Rev., Age, Author, and Last log entry. The 'env.sh' file is highlighted, showing its revision history. Below the table, there are options to filter files by tag and a link to download the directory as a tarball or zip archive.

File	Rev.	Age	Author	Last log entry
Parent Directory				
Analysis/				
CaloDist/				
Clustering/				
Pflow/				
TrackDist/				
Tracking/				
doc/				
examples/				
examples_LDC/				
src/				
GNUmakefile	1.1.1.1	4 months	aplin	Initial version
env.sh	1.1	2 weeks	tkraemer	Environment script for building MarlinReco as a collection of packages together...

The screenshot shows the MarlinReco documentation page. It includes a navigation sidebar and a main content area with the title 'MarlinReco: A Marlin based Reconstruction Package for the ILC'. The page provides an introduction to the package, its dependencies, and installation instructions. It also includes a table of contents and a list of authors.

Software Portal:  
**http://ilcsoft.desy.de**  
[aka: http://www-flc.desy.de/ilcsoft]

# Summary/Outlook

- **MarlinReco is a reconstruction toolkit for the ILC**
  - (almost) complete set of standard reconstruction
  - cheaters for cross checks (and surrogates)
  - **flexible**: can be used for other detector concepts
  - **extensible**: use/combine with other processors
- v00-01 released
  - plans to extend functionality exist
  - your input is welcome !

MarlinReco is a tool for answering the questions raised by the detector concept studies  
-> use it now !



Backup slides...

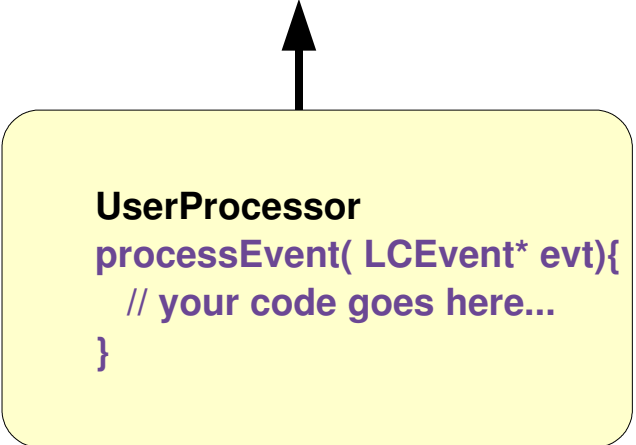


# Marlin Processor

- provides main **user callbacks**
- has **own set of input parameters**
  - int, float, string (single and arrays)
  - parameter description
- naturally modularizes the application
- **order of processors is defined via steering file:**
  - easy to exchange one or several modules w/o recompiling
  - can run the same processor with different parameter set in one job
- **processor task can be as simple as creating one histogram or as complex as track finding and fitting in the central tracker**

```
marlin::Processor
init()
processRunHeader(LCRunHeader* run)
processEvent( LCEvent* evt)
check( LCEvent* evt)
end()
```

```
UserProcessor
processEvent( LCEvent* evt){
    // your code goes here...
}
```

A diagram showing inheritance. A box containing the UserProcessor code is positioned below a box containing the marlin::Processor code. A black arrow points upwards from the UserProcessor box to the marlin::Processor box, indicating that UserProcessor inherits from marlin::Processor.



# XML files configure a Marlin Application

```
- <marlin>
- <execute>
  <processor name="MyAIDAProcessor"/>
  <processor name="MyEventSelection"/>
  - <if condition="MyEventSelection">
    <group name="Tracking"/>
    <processor name="MyClustering"/>
    <processor name="MyPFlow"/>
    <processor name="MyLCIOOutputProcessor"/>
  </if>
</execute>
- <global>
  <parameter name="LCIOInputFiles"> simjob.slcio </parameter>
  <parameter name="MaxRecordNumber" value="5001"/>
  <parameter name="SupressCheck" value="false"/>
</global>
- <processor name="MyLCIOOutputProcessor" type="LCIOOutputProcessor">
  <parameter name="LCIOOutputFile" type="string">outputfile.slcio </parameter>
  <parameter name="LCIOWriteMode" type="string">WRITE_NEW</parameter>
</processor>
- <group name="Tracking">
  <parameter name="NTPCLayers" value="200"/>
  <processor name="MyTrackfinder" type="Trackfinder"/>
  - <processor name="MyTrackfitter" type="Trackfitter">
    <parameter name="Algorithm" value="DAF"/>
  </processor>
</group>
<!-- ... -->
</marlin>
```

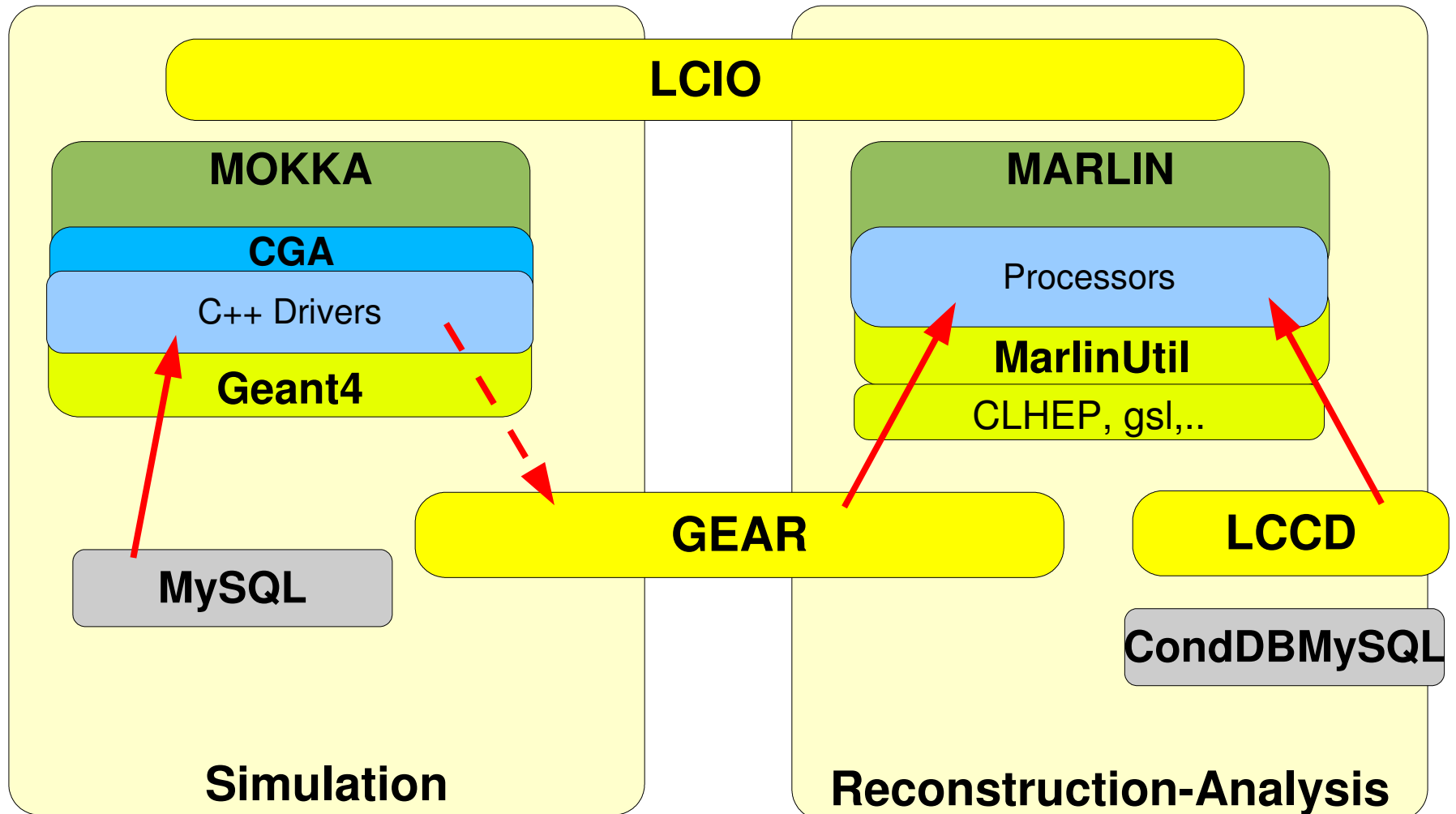
- Active processors defined in <execute>...</execute> section
- Logical conditions from booleans set by preceding processors

- Parameters: global and per processor
- Parameters defined as content of <parameter/> tag or as its value attribute

- Processors can be enclosed by <group/> tag
- Parameters in <group/> joined by all processors



# LDC simulation framework



# Marlin on the web

Frank Gaede, ECFA ILC Workshop, Vienna, Nov 14-17, 2005

MARLIN homepage - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://ilcsoft.desy.de/marlin/

simulation/geant4 LCIO Linux DESY IT Group LEO English/Ger... Google MyHome The 2005 Internat...

## Modular Analysis & Reconstruction for the LINear Collider

### Releases

v00-08 has been released and is available for [download](#).  
Marlin can now optionally be linked against [LCCD](#) to provide easy access to conditions data.  
[documentation](#) has been improved.

### Download

All tagged versions and the current HEAD of the repository can be downloaded from the [CVS](#).

### Documentation

[Current API documentation](#).

### Talks

LCIO & Marlin ([pdf](#)) - talk given at the DESY Simulation WS 2004.

Last modified: Fri Mar 11 16:01:59 2005

by [Frank.Gaede@desy.de](mailto:Frank.Gaede@desy.de)

Done

<http://ilcsoft.desy.de/marlin>

latest release: v00-09-02

Marlin: <a href="http://www.desy.de/~gaede/marlin">Marlin</a> (v00-08) - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://ilcsoft.desy.de/marlin/v00-08/doc/html/index.html

simulation/geant4 LCIO Linux DESY IT Group LEO English/Ger... Google MyHome The 2005 Internat...

## Marlin (v00-08)

Marlin [Modular Analysis and Reconstruction for the LINear collider] is a simple modular application framework for analysis and reconstruction code based on LCIO.

### Overview

Marlin is a simple modular application framework for analysis and reconstruction code based on LCIO. As a lot of different groups are involved in the development of reconstruction and analysis code based on LCIO, it is necessary to have distributed development of modules and combine existing modules as needed in a larger application. Marlin provides a simple framework for this. It defines a set of callbacks that the user can implement in their subclasses. A steering file is used to configure the application. These are then called for every event using the LCEvent as container for input and output data in the application.

```
graph TD
    MyInput[MyInput.slcio] --> LCEvent[LCEvent]
    LCEvent --> collection0[collection0]
    collection0 --> marlin_main[marlin::main]
    marlin_main --> Processor0[Processor0]
    marlin_main --> Processor1[Processor1]
    marlin_main --> Processor2[Processor2]
    marlin_main --> ProcessorN[ProcessorN]
    marlin_main --> OutputProcessor[OutputProcessor]
```

### Installation

The installation of Marlin is described in the [README](#).

### Running Marlin

After having installed Marlin you have to write your own marlin::Processor(s) subclass that performs the computation. This is fairly straight forward and Marlin provides an example in [/examples/mymarlin](#) that can serve as a template for your own projects.

Note: there is no need to write a main program as this is provided by Marlin. Existing Processors are automatically registered with Marlin provided one instance exists in the library as described in the [README](#).

### Steering files

Done