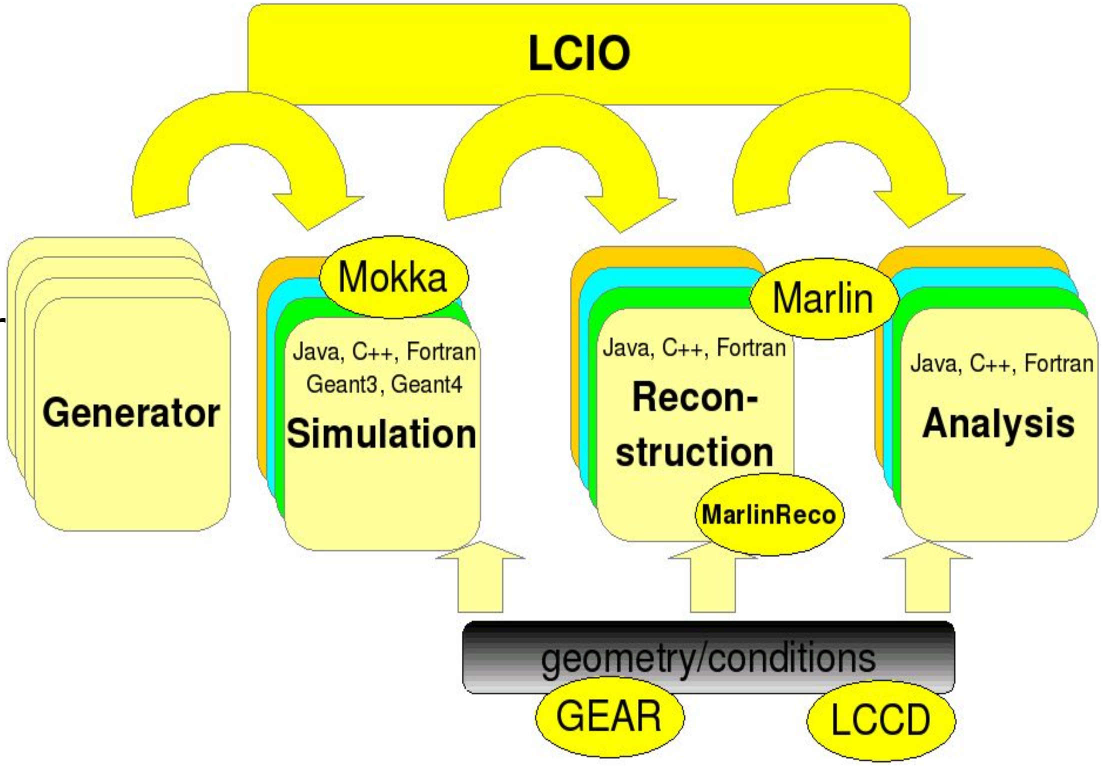
ILC Core Software

in Europe

Frank Gaede
DESY
ILC ECFA Workshop,
Valencia, 6-10 November 2006

Outline

- Core software tools - status and latest developments
  - LCIO
    - data model & persistency
  - Marlin
    - C++ application framework
  - MarlinReco
    - Marlin based reconstruction
  - LCCD
    - conditions data toolkit
  - GEAR
    - geometry description
  - LCGO
    - a proposal for a new geometry package:
- Summary



LCIO overview

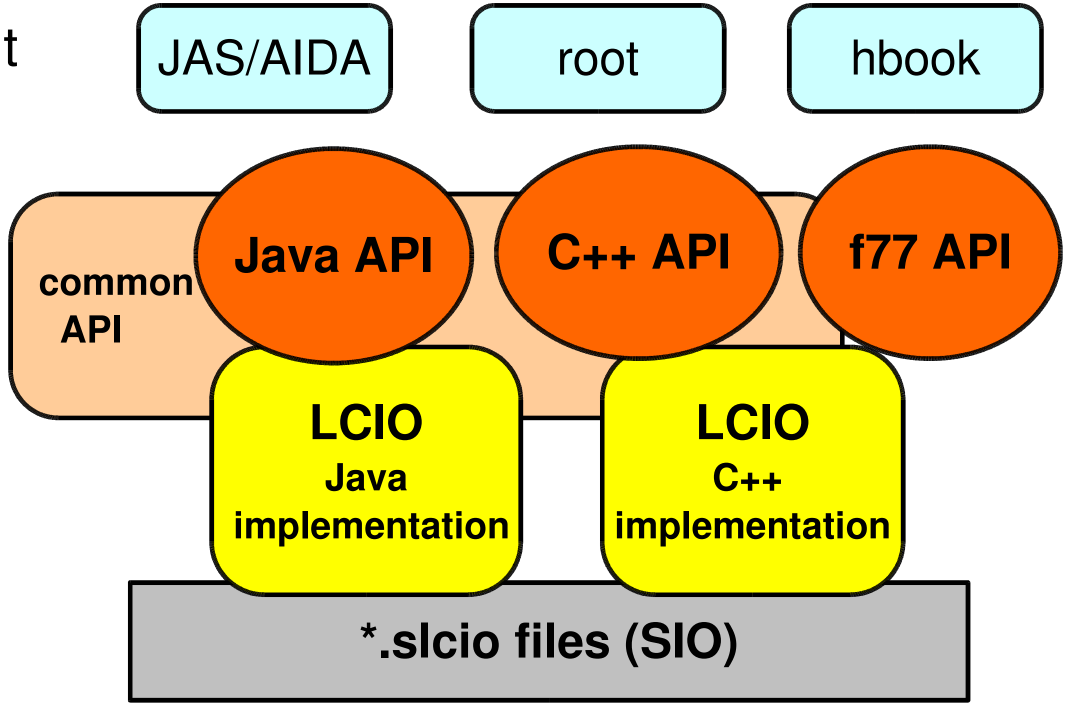
- DESY and SLAC joined project:
  - provide common basis for ILC software
- Features:
  - Java, C++ and f77 (!) API
  - extensible data model for current and future simulation and testbeam studies
  - user code separated from concrete data format
  - no dependency on other frameworks

simple & lightweight

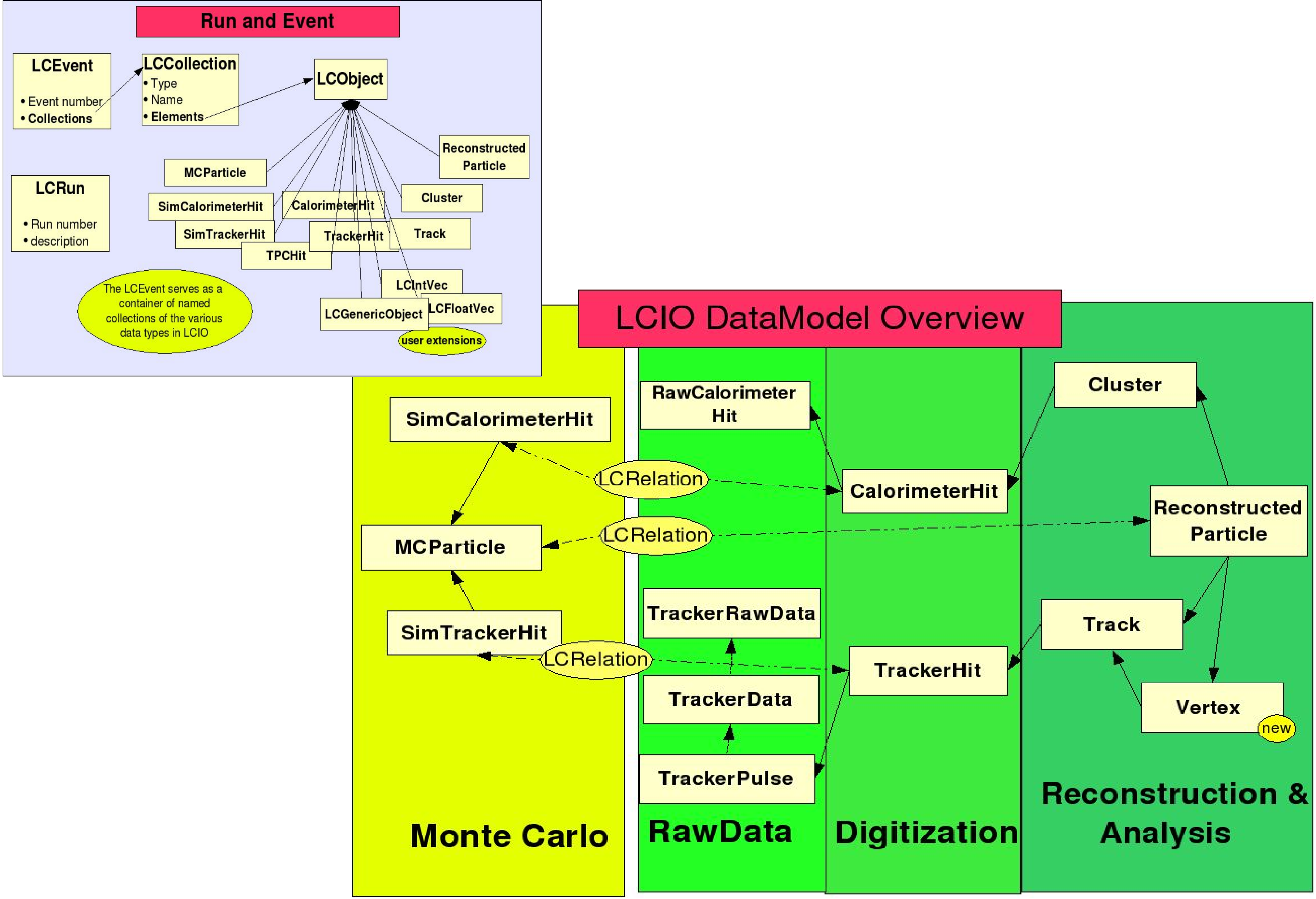
current release: **v01-07**

now de facto standard persistency & datamodel for ILC software

SW-Architecture



LCIO data model



Run and Event

LCEvent
- Event number
- Collections

LCCollection
- Type
- Name
- Elements

LCObject

LCRun
- Run number
- description

MCParticle

SimCalorimeterHit

SimTrackerHit

TPCHit

CalorimeterHit

TrackerHit

Cluster

Track

Reconstructed Particle

LCIntVec

LCFloatVec

LCGenericObject

user extensions

The LCEvent serves as a container of named collections of the various data types in LCIO

LCIO DataModel Overview

Monte Carlo

SimCalorimeterHit

MCParticle

SimTrackerHit

RawData

RawCalorimeter Hit

TrackerRawData

TrackerData

TrackerPulse

Digitization

CalorimeterHit

TrackerHit

Reconstruction & Analysis

Cluster

Reconstructed Particle

Track

Vertex

new

LCRelation

LCRelation

LCRelation

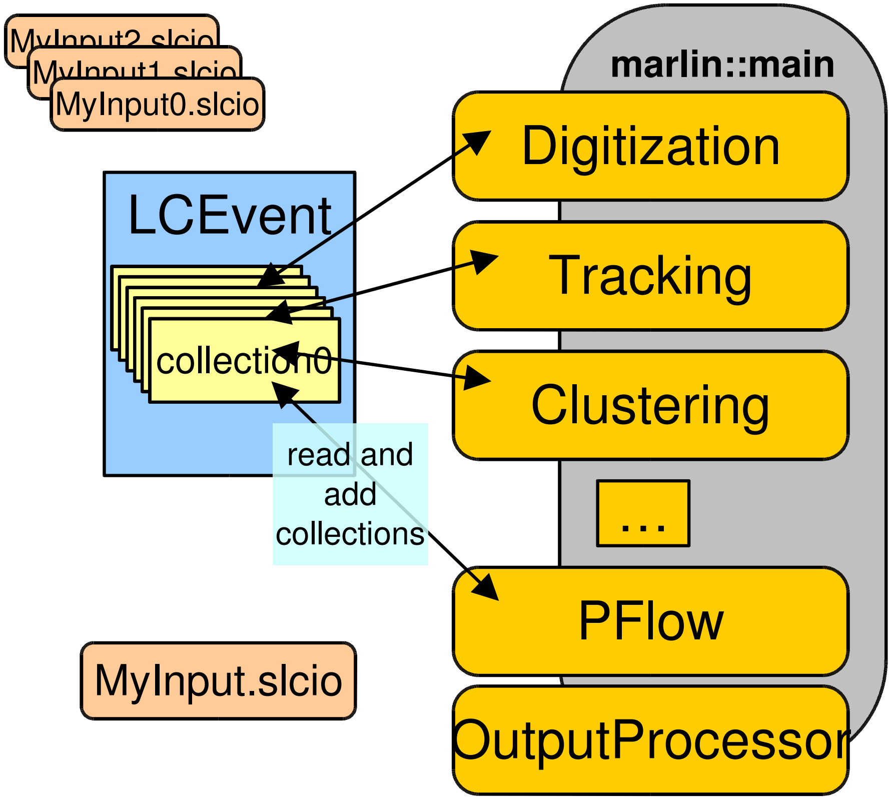
LCIO Status

- the LCIO event data model is fairly complete & flexible
- however it is adapted and extended as needed by the community
  - maintaining downward compatibility
  - with international discussion and agreement
- examples:
- new raw data classes for prototypes
  - TPC uses TrackerRawData, TrackerData, TrackerPulse
  - also to be used for vertex prototypes
- introduction of a new **Vertex** class in LCIO
  - originally proposed by LCFI group
  - see discussion @ http://forum.linearcollider.org/
- test release v01-08-vtx
- -> will become production release v01-08 soon

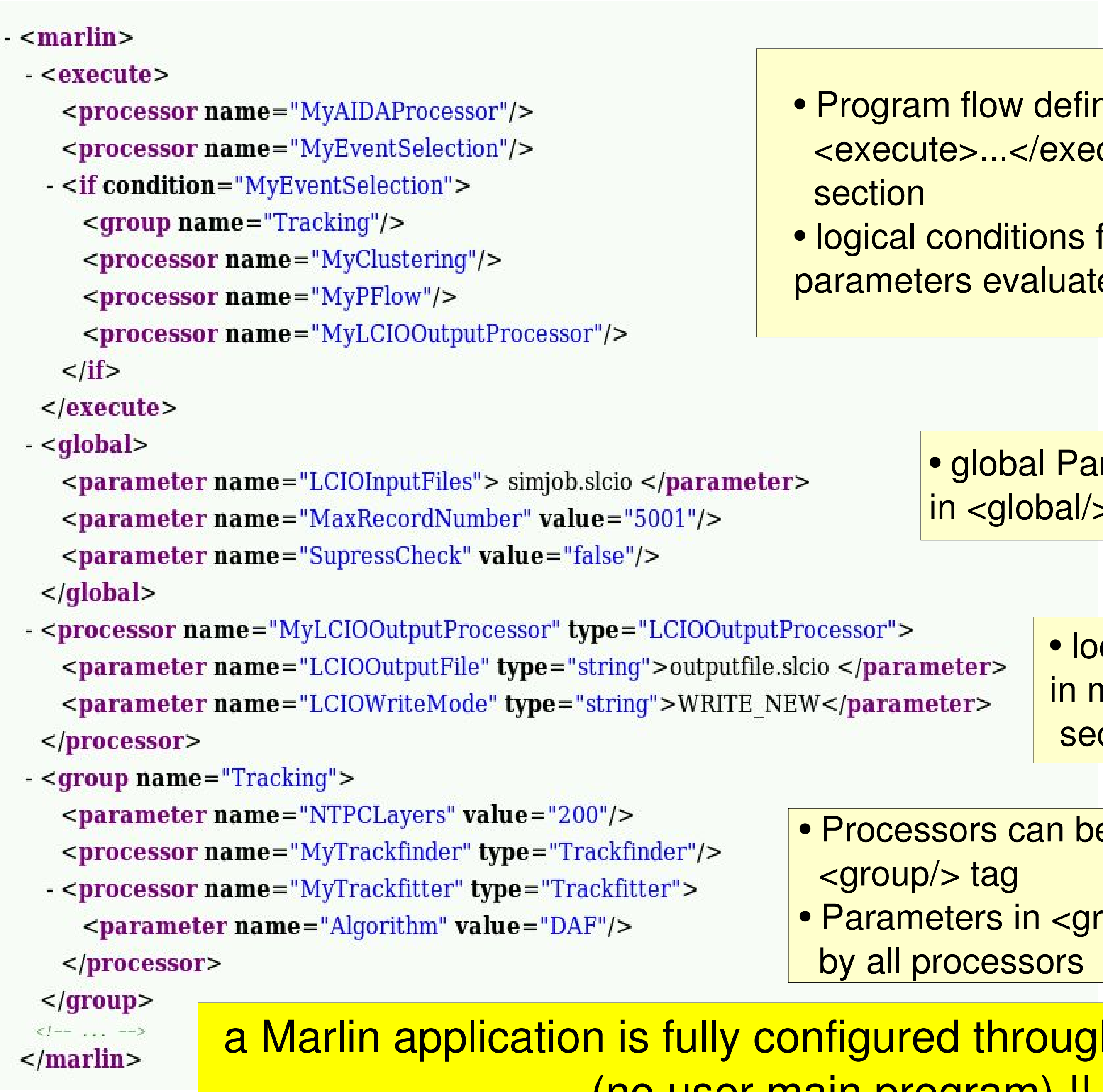
Marlin

Modular**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
  - **Plug&Play** of processors



Marlin – XML steering files



```
- <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>
```

- Program flow defined in <execute>...</execute> section
- logical conditions from parameters evaluated at runtime

- global Parameters defined in <global/> section

- local Parameters defined in mandatory <parameter/> section

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

a Marlin application is fully configured through the steering files
(no user main program) !!

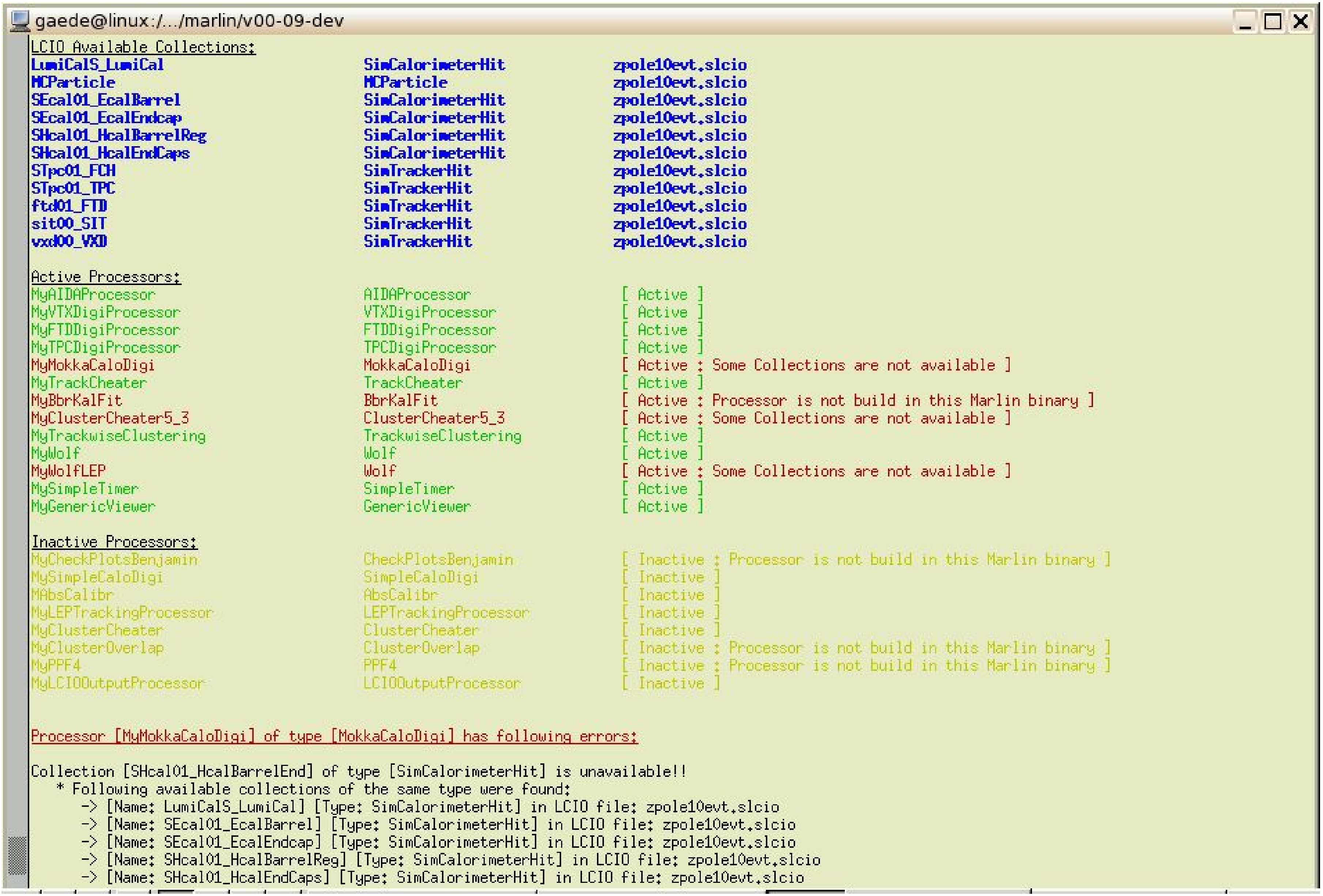
Marlin Status and Plans

- current version: v00-09-05:
  - made compatible with CLHEP 2.x
  - made compatible with RAIDA (root impl. of AIDA)
  - split outputfiles wrt. size: ttbar_000.slcio, ttbar_001.slcio,...
- new version (soon)

J.Engels (EUDET)

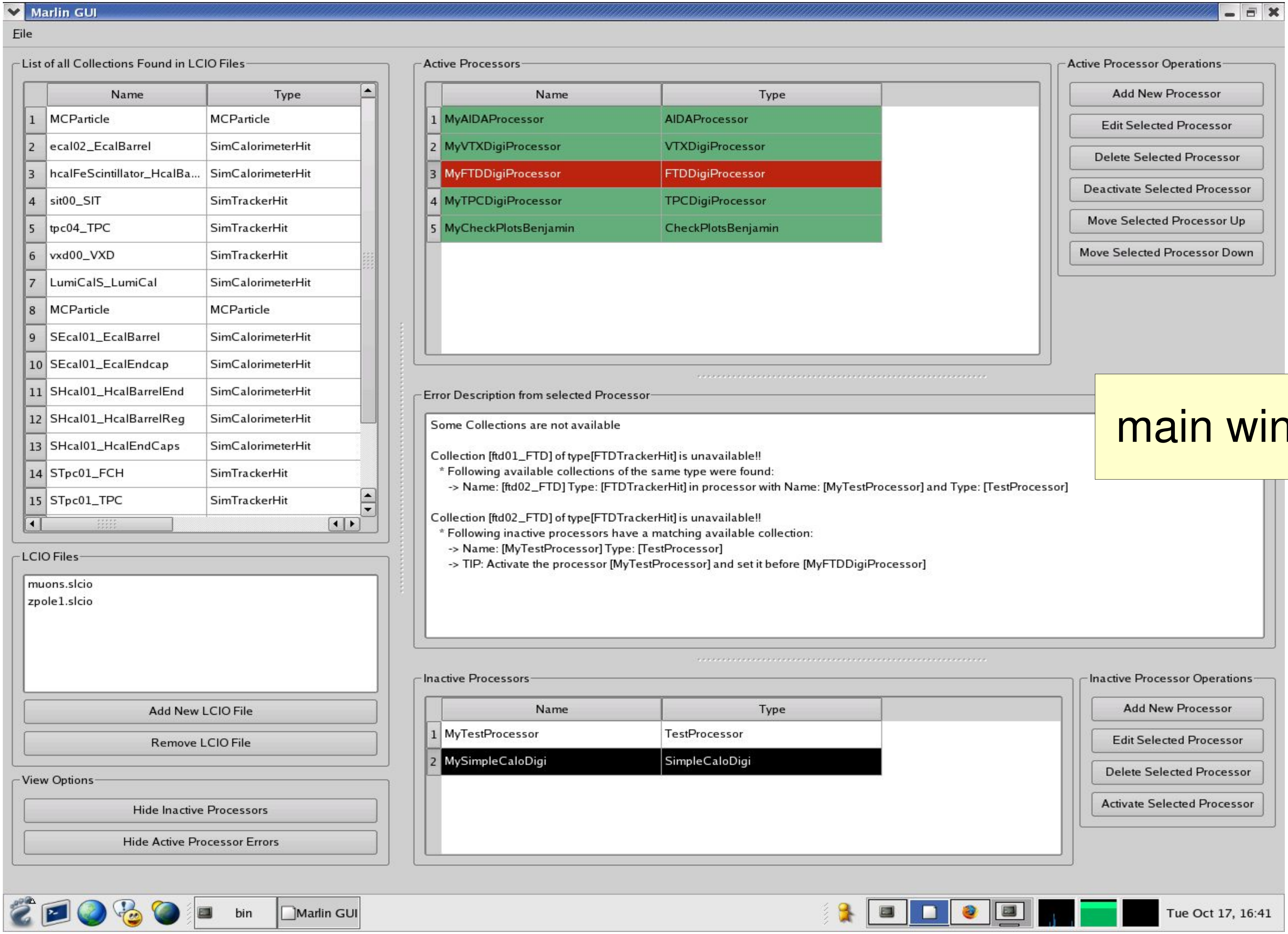
  - Marlin -c mysteer.xml
    - consistency check and prints error messages/hints
  - Marlin -f oldsteer.xml newsteer.xml
    - adds LCIO collection type information
  - Processor::addInput/OutputCollection() to be called by users/authors of marlin processors
  - MarlinGUI:
    - interactive creation/modification of steering files
    - based on QT4

Example: Marlin -c steer.xml



```
LCIO Available Collections:
LumiCalS_LumiCal                        SimCalorimeterHit             zpole10evt.slcio
MCParticle                              MCParticle                    zpole10evt.slcio
SEcal01_EcalBarrel                      SimCalorimeterHit             zpole10evt.slcio
SEcal01_EcalEndcap                      SimCalorimeterHit             zpole10evt.slcio
SHcal01_HcalBarrelReg                   SimCalorimeterHit             zpole10evt.slcio
SHcal01_HcalEndCaps                     SimCalorimeterHit             zpole10evt.slcio
STpc01_FCH                              SimTrackerHit                 zpole10evt.slcio
STpc01_TPC                              SimTrackerHit                 zpole10evt.slcio
ftd01_FTD                               SimTrackerHit                 zpole10evt.slcio
sit00_SIT                               SimTrackerHit                 zpole10evt.slcio
vxd00_VXD                               SimTrackerHit                 zpole10evt.slcio

Active Processors:
MyAIDAProcessor                         AIDAProcessor                  [ Active ]
MyVTXDigiProcessor                      VTXDigiProcessor               [ Active ]
MyFTDDigiProcessor                      FTDDigiProcessor               [ Active ]
MyTPCDigiProcessor                      TPCDigiProcessor               [ Active ]
MyMokkaCaloDigi                         MokkaCaloDigi                  [ Active : Some Collections are not available ]
MyTrackCheater                          TrackCheater                   [ Active ]
MyBbrKalFit                             BbrKalFit                      [ Active : Processor is not build in this Marlin binary ]
MyClusterCheater5_3                     ClusterCheater5_3              [ Active : Some Collections are not available ]
MyTrackwiseClustering                   TrackwiseClustering            [ Active ]
MyWolf                                  Wolf                           [ Active ]
MyWolfLEP                               Wolf                           [ Active : Some Collections are not available ]
MySimpleTimer                           SimpleTimer                    [ Active ]
MyGenericViewer                         GenericViewer                  [ Active ]

Inactive Processors:
MyCheckPlotsBenjamin                    CheckPlotsBenjamin             [ Inactive : Processor is not build in this Marlin binary ]
MySimpleCaloDigi                        SimpleCaloDigi                 [ Inactive ]
MAbsCalibr                              AbsCalibr                      [ Inactive ]
MyLEPTrackingProcessor                  LEPTrackingProcessor           [ Inactive ]
MyClusterCheater                        ClusterCheater                 [ Inactive ]
MyClusterOverlap                        ClusterOverlap                 [ Inactive : Processor is not build in this Marlin binary ]
MyPPF4                                  PPF4                           [ Inactive : Processor is not build in this Marlin binary ]
MyLCIOOutputProcessor                   LCIOOutputProcessor            [ Inactive ]


Processor [MyMokkaCaloDigi] of type [MokkaCaloDigi] has following errors:

Collection [SHcal01_HcalBarrelEnd] of type [SimCalorimeterHit] is unavailable!!
   * Following available collections of the same type were found:
      -> [Name: LumiCalS_LumiCal] [Type: SimCalorimeterHit] in LCIO file: zpole10evt.slcio
      -> [Name: SEcal01_EcalBarrel] [Type: SimCalorimeterHit] in LCIO file: zpole10evt.slcio
      -> [Name: SEcal01_EcalEndcap] [Type: SimCalorimeterHit] in LCIO file: zpole10evt.slcio
      -> [Name: SHcal01_HcalBarrelReg] [Type: SimCalorimeterHit] in LCIO file: zpole10evt.slcio
      -> [Name: SHcal01_HcalEndCaps] [Type: SimCalorimeterHit] in LCIO file: zpole10evt.slcio
```

example: MarlinGUI



	Name	Type
1	MCParticle	MCParticle
2	ecal02_EcalBarrel	SimCalorimeterHit
3	hcalFeScintillator_HcalBa...	SimCalorimeterHit
4	sit00_SIT	SimTrackerHit
5	tpc04_TPC	SimTrackerHit
6	vxd00_VXD	SimTrackerHit
7	LumiCalS_LumiCal	SimCalorimeterHit
8	MCParticle	MCParticle
9	SEcal01_EcalBarrel	SimCalorimeterHit
10	SEcal01_EcalEndcap	SimCalorimeterHit
11	SHcal01_HcalBarrelEnd	SimCalorimeterHit
12	SHcal01_HcalBarrelReg	SimCalorimeterHit
13	SHcal01_HcalEndCaps	SimCalorimeterHit
14	STpc01_FCH	SimTrackerHit
15	STpc01_TPC	SimTrackerHit

muons.slcio
zpole1.slcio

	Name	Type
1	MyAIDAProcessor	AIDAProcessor
2	MyVTXDigiProcessor	VTXDigiProcessor
3	MyFTDDigiProcessor	FTDDigiProcessor
4	MyTPCDigiProcessor	TPCDigiProcessor
5	MyCheckPlotsBenjamin	CheckPlotsBenjamin

Active Processor Operations
Add New Processor
Edit Selected Processor
Delete Selected Processor
Deactivate Selected Processor
Move Selected Processor Up
Move Selected Processor Down
Error Description from selected Processor

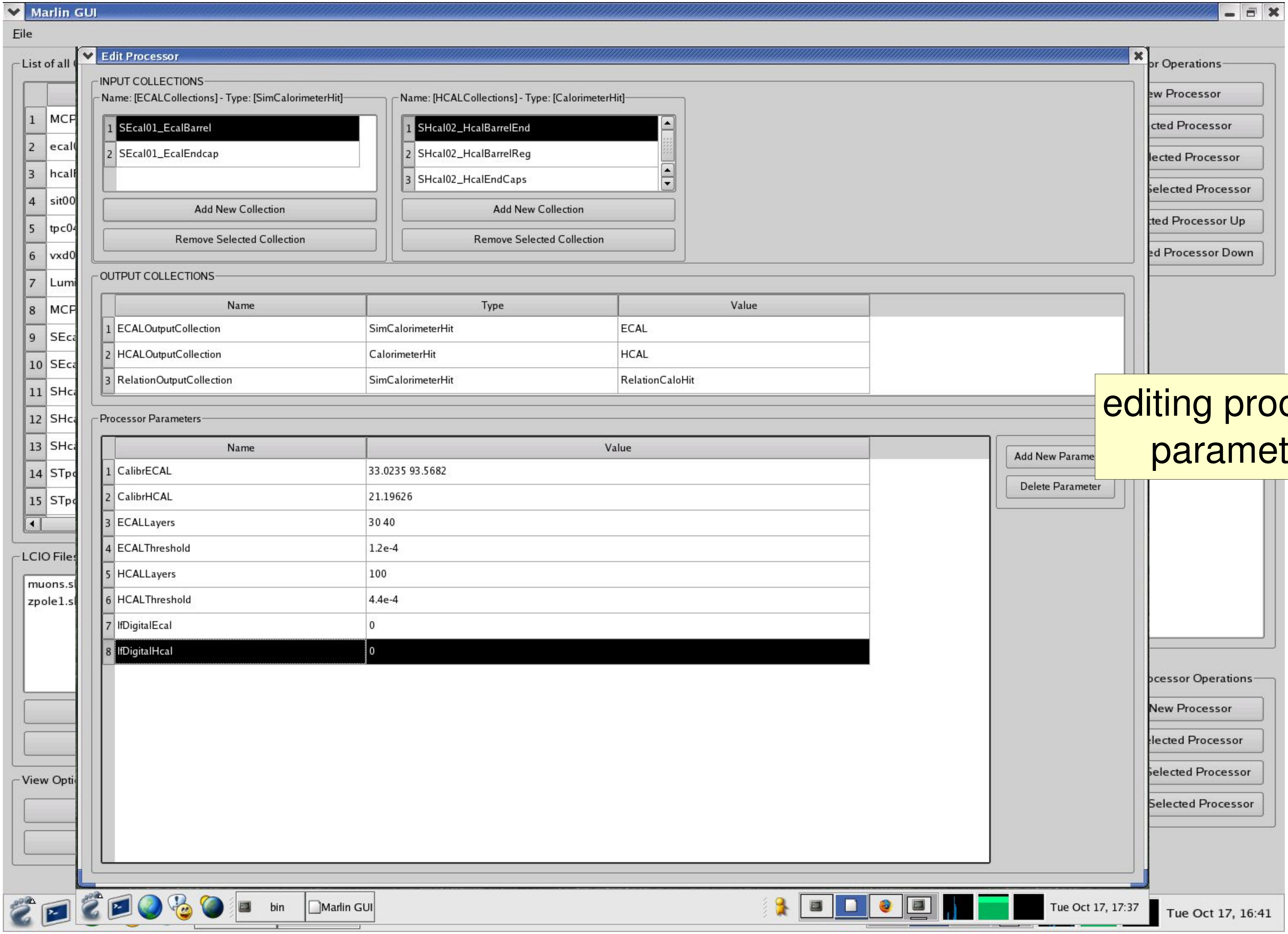
```
Some Collections are not available

Collection [ftd01_FTD] of type[FTDTrackerHit] is unavailable!!
  * Following available collections of the same type were found:
    -> Name: [ftd02_FTD] Type: [FTDTrackerHit] in processor with Name: [MyTestProcessor] and Type: [TestProcessor]

Collection [ftd02_FTD] of type[FTDTrackerHit] is unavailable!!
  * Following inactive processors have a matching available collection:
    -> Name: [MyTestProcessor] Type: [TestProcessor]
    -> TIP: Activate the processor [MyTestProcessor] and set it before [MyFTDDigiProcessor]
```

	Name	Type
1	MyTestProcessor	TestProcessor
2	MySimpleCaloDigi	SimpleCaloDigi

Inactive Processor Operations
Add New Processor
Edit Selected Processor
Delete Selected Processor
Activate Selected Processor
bin
Marlin GUI
Tue Oct 17, 16:41

main window

example: MarlinGUI



1	MCP
2	ecal(
3	hcalF
4	sit00
5	tpc04
6	vxd0
7	Lumi
8	MCP
9	SEca
10	SEca
11	SHca
12	SHca
13	SHca
14	STpc
15	STpc

INPUT COLLECTIONS

Name: [ECALCollections] - Type: [SimCalorimeterHit]

1 SEcal01_EcalBarrel
2 SEcal01_EcalEndcap

Name: [HCALCollections] - Type: [CalorimeterHit]

1 SHcal02_HcalBarrelEnd
2 SHcal02_HcalBarrelReg
3 SHcal02_HcalEndCaps

OUTPUT COLLECTIONS

	Name	Type	Value
1	ECALOutputCollection	SimCalorimeterHit	ECAL
2	HCALOutputCollection	CalorimeterHit	HCAL
3	RelationOutputCollection	SimCalorimeterHit	RelationCaloHit

Processor Parameters

	Name	Value
1	CalibrECAL	33.0235 93.5682
2	CalibrHCAL	21.19626
3	ECALLayers	30 40
4	ECALThreshold	1.2e-4
5	HCALLayers	100
6	HCALThreshold	4.4e-4
7	IfDigitalEcal	0
8	IfDigitalHcal	0

editing processor parameters

MarlinReco

MarlinReco is a Marlin based **toolkit** providing reconstruction algorithms for the detector concept studies - packages:

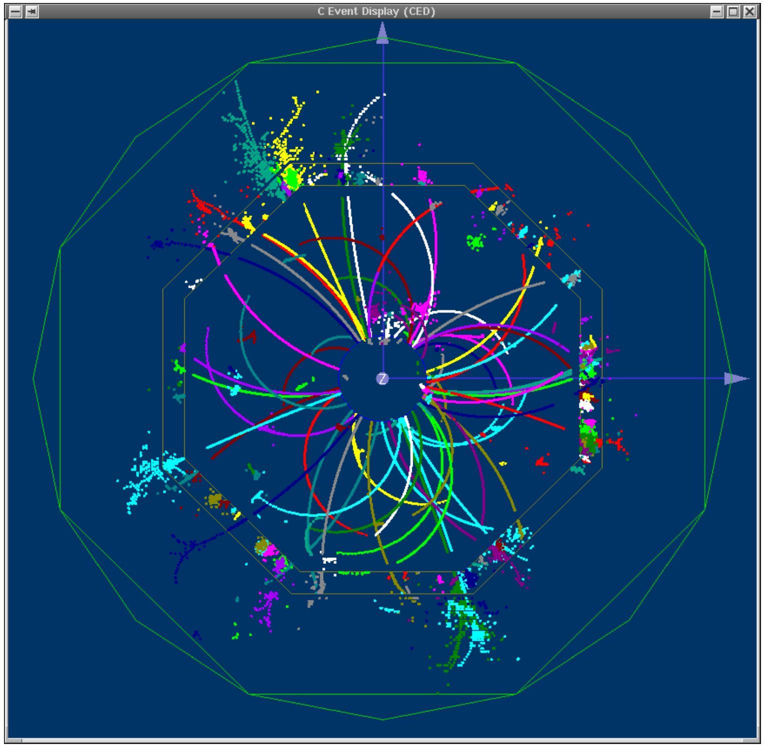
- **TrackDigi**
  - TPCDigi
  - new VTXDigi
- **CaloDigi**
  - LDCCaloDigi
- **Tracking**
  - LEPTracking (f77)
  - new VTXTracking
  - TrackCheater
- **Clustering**
  - TrackwiseClustering
  - ClusterCheater
- **Pflow**
  - Wolf
- **Analysis**
  - EventShapes
  - SatoruJetFinder

- MarlinReco supports distributed development of reconstruction code
- packages can be seamlessly integrated with other packages, e.g.
  - PandoraPFA (M.Thomson)
  - ZVTop (S.Hillert/LCFI)

will become part of MarlinReco cvs repository !

MarlinReco support packages

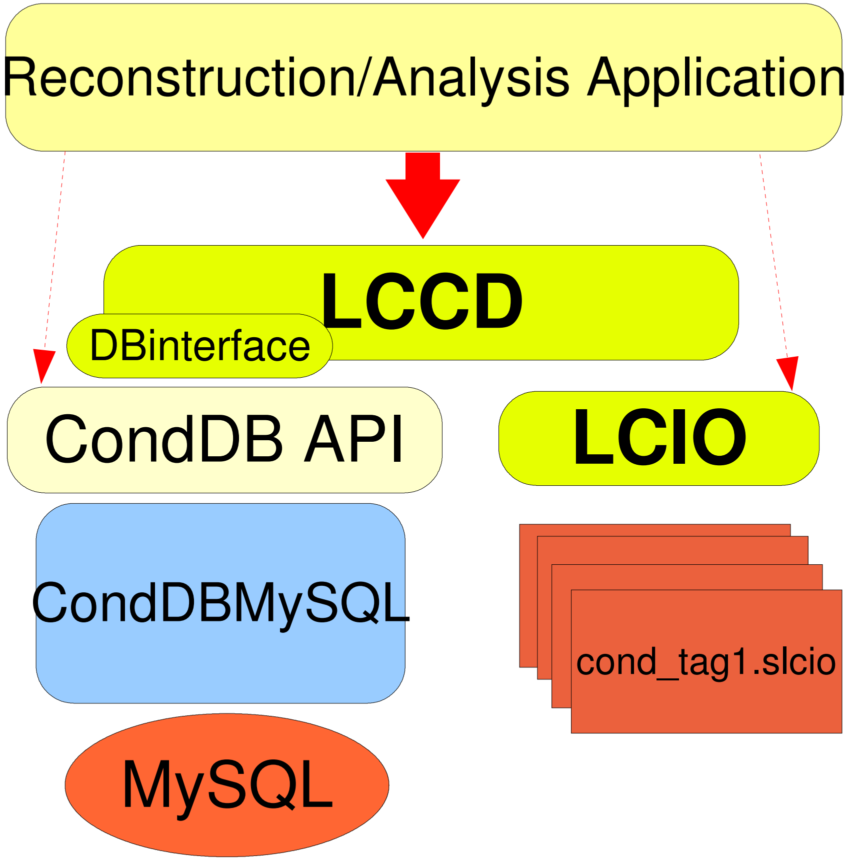
- MarlinUtil (O. Wendt, T.Kraemer)
  - Utility and Helper classes
    - helix fitter, cluster shapes,...
    - trajectory class / extrapolation new
- RAIDA (T.Kraemer)
    - AIDA root implementation
- CED (A. Zhelezov)
    - event display based on GLUT/ OpenGL
    - client server architecture
- CEDViewer
  - event display client processors
    - CEDViewer, GenericViewer



LCCD

Linear **C**ollider **C**onditions **D**ata Toolkit

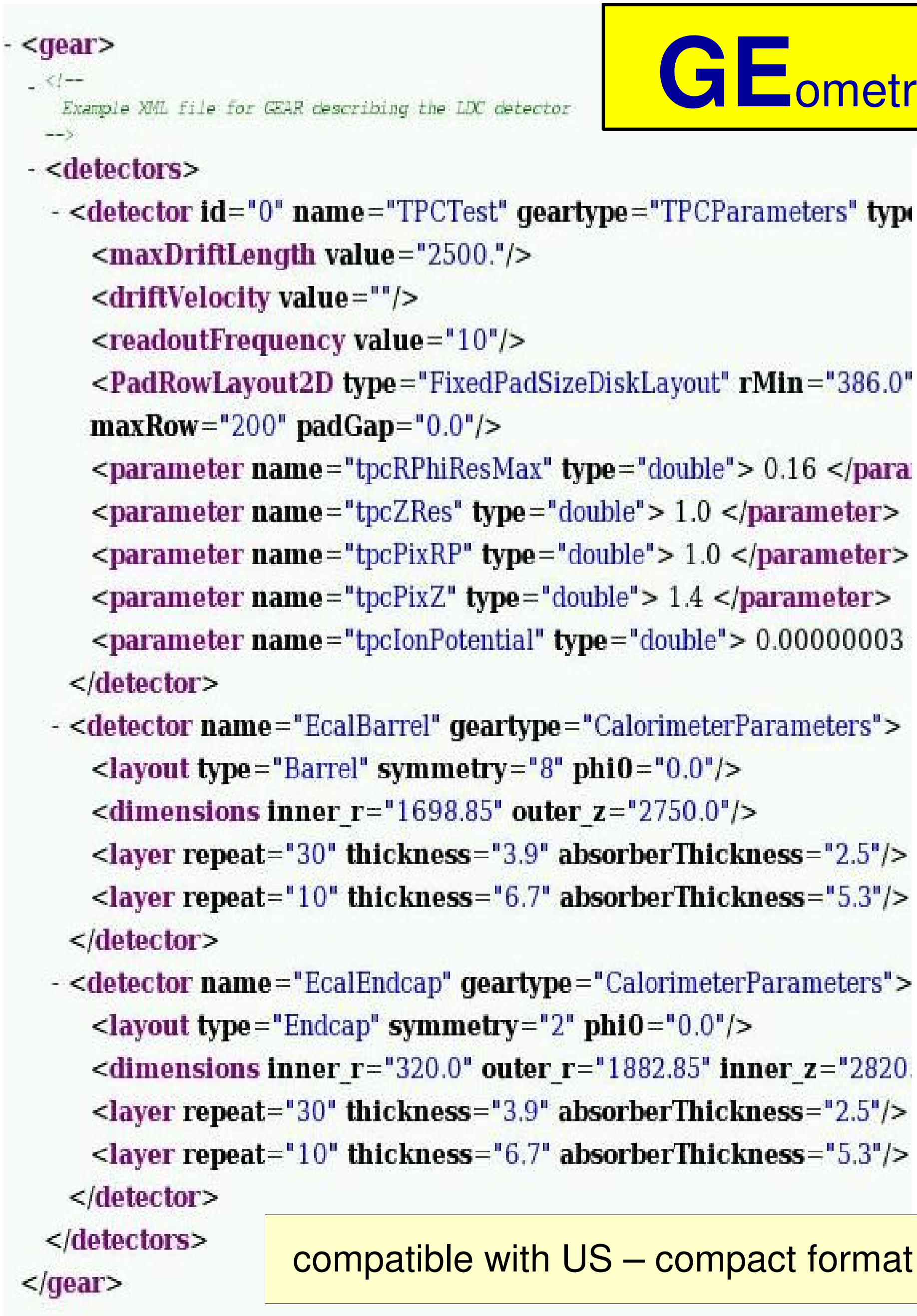
- Reading conditions data
  - from conditions database
  - from simple LCIO file
  - from LCIO data stream
  - from dedicated LCIO-DB file
- Writing conditions data
- tag conditions data
- Browse the conditions database
  - through creation of LCIO files
    - vertically (all versions for timestamp)
    - horizontally (all versions for tag)



LCCD is used by Calice for the conditions data of the ongoing testbeam studies

Gear

GEometry API for Reconstruction



```
- <gear>
  - <!--
     Example XML file for GEAR describing the LDC detector
    -->
  - <detectors>
    - <detector id="0" name="TPCTest" geartype="TPCParameters" typ
        <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 </para
        <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
      </detector>
    - <detector name="EcalBarrel" geartype="CalorimeterParameters">
        <layout type="Barrel" symmetry="8" phi0="0.0"/>
        <dimensions inner_r="1698.85" outer_z="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.
        <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

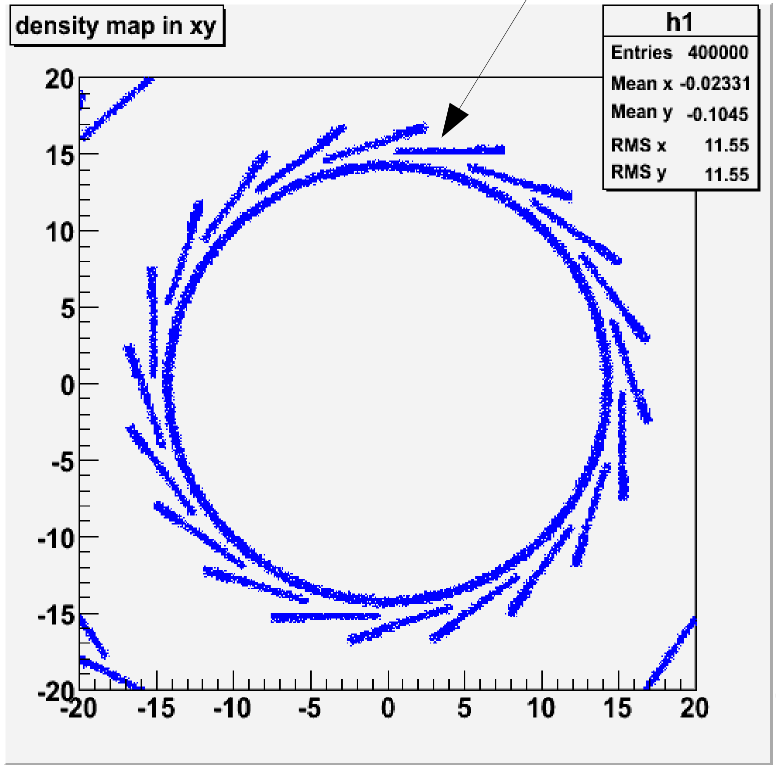
- well defined geometry definition for reconstruction that
  - is flexible w.r.t different detector concepts
  - has high level information needed for reconstruction
  - provides access to material properties
- abstract interface (a la LCIO)
  - concrete implementation based on XML files
  - and Mokka-CGA

Gear status

- version v00-03
  - TPC, Hcal, Ecal and VXD (new) interfaces defined and implemented
  - arbitrary named (user) parameters
  - write xml files from parameters in memory
  - tool to merge files: gearmerge
  - description of TPC prototypes (rectangular pad plane)
- GearCGA - material properties
  - (implemented by G. Mussat, LLR)
- related work: MokkaGear

• exact geant4 material & field information at runtime !
• performance ?
• practical issues (linking g4) ?

example: density in r-phi from GearPointProperties

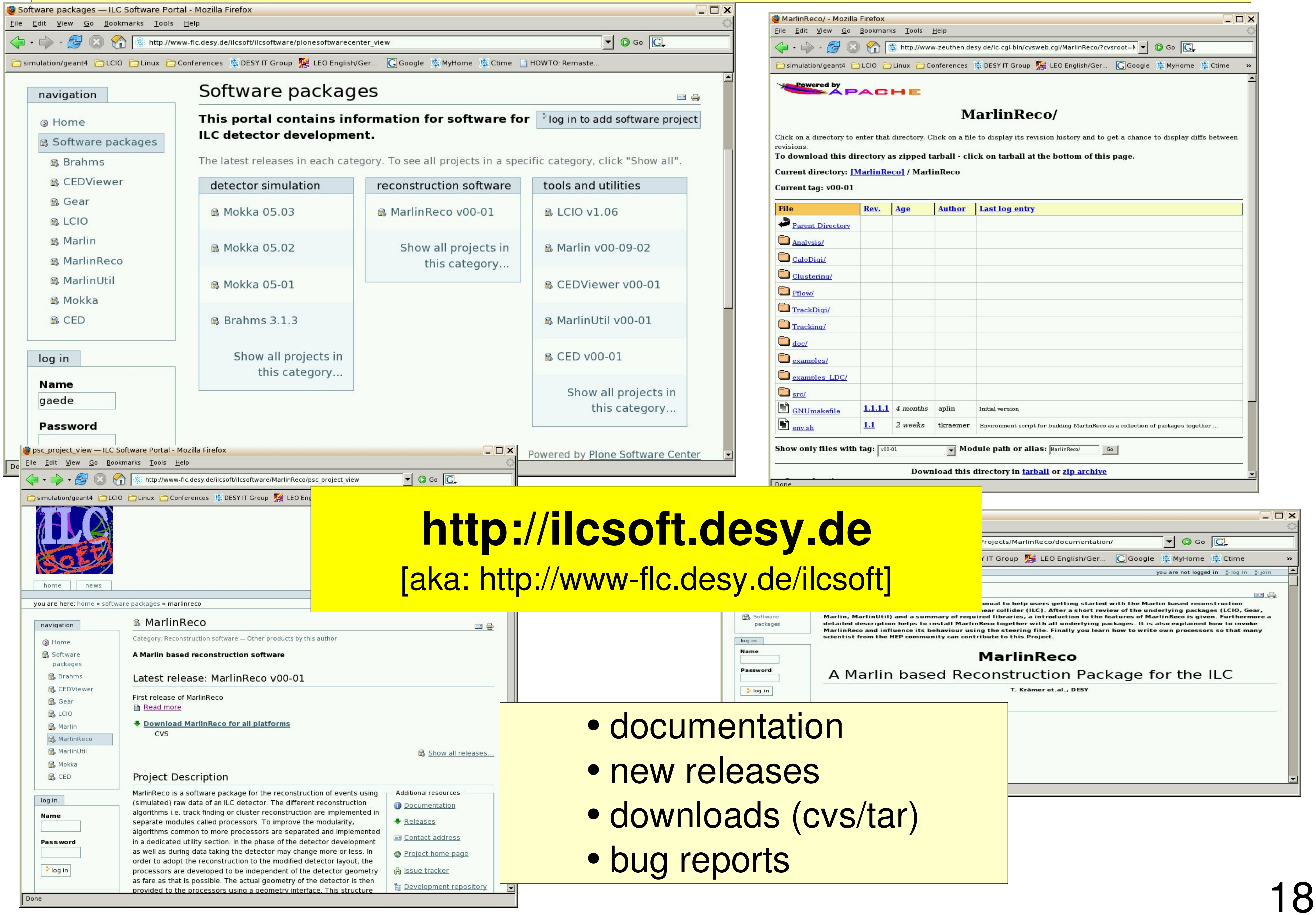


MokkaGear

- extension to Mokka (R.Lippe – Diploma Thesis)
- extract geometry information in drivers when detector is built
- use Gear to create XML files for reconstruction
- currently implemented:
  - TPC (tpc04), Ecal (ecal02) and Hcal (hcal04)
- **released with Mokka 6.1**
- optional feature
  - only if Gear is installed and included

aim: have only one source of of information
for describing the detector geometry !

ILC software portal



Software packages — ILC Software Portal - Mozilla Firefox

http://www-flc.desy.de/ilcsoft/ilcsoftware/plonesoftwarecenter_view

File Edit View Go Bookmarks Tools Help
Go
simulation/geant4 LCIO Linux Conferences DESY IT Group LEO English/Ger... Google MyHome Ctime HOWTO: Remaste...
navigation
Home
Software packages
Brahms
CEDViewer
Gear
LCIO
Marlin
MarlinReco
MarlinUtil
Mokka
CED
log in

Name: gaede

Password

Software packages

This portal contains information for software for ILC detector development.

log in to add software project

The latest releases in each category. To see all projects in a specific category, click "Show all".

detector simulation	reconstruction software	tools and utilities
Mokka 05.03	MarlinReco v00-01	LCIO v1.06
Mokka 05.02	Show all projects in this category...	Marlin v00-09-02
Mokka 05-01		CEDViewer v00-01
Brahms 3.1.3		MarlinUtil v00-01
Show all projects in this category...		CED v00-01
		Show all projects in this category...

Powered by Plone Software Center

MarlinReco/ - Mozilla Firefox

http://www-zeuthen.desy.de/lc-cgi-bin/cvsweb.cgi/MarlinReco/?cvsroot=

File Edit View Go Bookmarks Tools Help
Go
simulation/geant4 LCIO Linux Conferences DESY IT Group LEO English/Ger... Google MyHome Ctime
Powered by APACHE

MarlinReco/

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display diffs between revisions.

To download this directory as zipped tarball - click on tarball at the bottom of this page.

Current directory: [MarlinReco] / MarlinReco

Current tag: v00-01

File	Rev.	Age	Author	Last log entry
Parent Directory				
Analysis/				
CaloDigi/				
Clustering/				
Pflow/				
TrackDigi/				
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 ...

Show only files with tag: v00-01 **Module path or alias:** MarlinReco/ Go

Download this directory in tarball or zip archive

Done

psc_project_view — ILC Software Portal - Mozilla Firefox

http://www-flc.desy.de/ilcsoft/ilcsoftware/MarlinReco/psc_project_view

File Edit View Go Bookmarks Tools Help
Go
simulation/geant4 LCIO Linux Conferences DESY IT Group LEO Eng
ILC SOFT
home
news

you are here: home » software packages » marlinreco

MarlinReco

Category: Reconstruction software — Other products by this author

A Marlin based reconstruction software

Latest release: MarlinReco v00-01

First release of MarlinReco

Read more

Download MarlinReco for all platforms

CVS

Show all releases...

Project Description

MarlinReco is a software package for the reconstruction of events using (simulated) raw data of an ILC detector. The different reconstruction algorithms i.e. track finding or cluster reconstruction are implemented in separate modules called processors. To improve the modularity, algorithms common to more processors are separated and implemented in a dedicated utility section. In the phase of the detector development as well as during data taking the detector may change more or less. In order to adopt the reconstruction to the modified detector layout, the processors are developed to be independent of the detector geometry as fare as that is possible. The actual geometry of the detector is then provided to the processors using a geometry interface. This structure

Additional resources

- Documentation
- Releases
- Contact address
- Project home page
- Issue tracker
- Development repository

Done

rojects/MarlinReco/documentation/

Go
IT Group LEO English/Ger... Google MyHome Ctime
you are not logged in log in join
Software packages
log in
Name
Password
log in

anual to help users getting started with the Marlin based reconstruction ear collider (ILC). After a short review of the underlying packages (LCIO, Gear, Marlin, MarlinUtil) and a summary of required libraries, a introduction to the features of MarlinReco is given. Furthermore a detailed description helps to install MarlinReco together with all underlying packages. It is also explained how to invoke MarlinReco and influence its behaviour using the steering file. Finally you learn how to write own processors so that many scientist from the HEP community can contribute to this Project.

MarlinReco

A Marlin based Reconstruction Package for the ILC

T. Krämer et.al., DESY

http://ilcsoft.desy.de

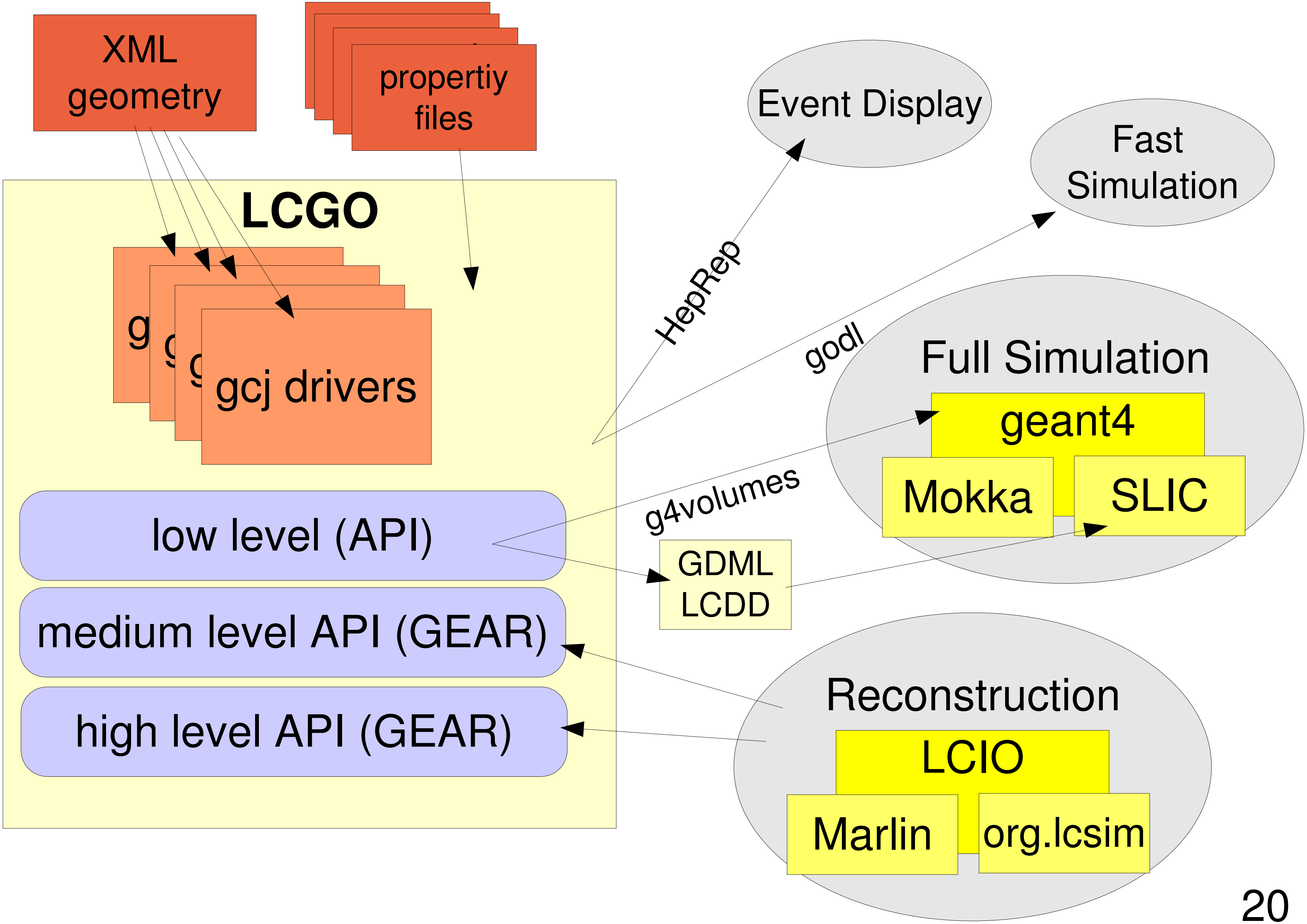
[aka: http://www-flc.desy.de/ilcsoft]

- documentation
- new releases
- downloads (cvs/tar)
- bug reports

A Common Geometry Toolkit

- **LCGO**: A common geometry toolkit to be used in all(?) ILC frameworks
  - SLAC-DESY project - initially
  - -> of course open for all collaborators
  - work just started – aiming for summer 2007
- requirements/goals for LCGO:
  - be at least as functional as existing systems (LCCD/SLIC, GEAR/Mokka)
  - enable smooth transition path from existing systems
  - encourage/increase interoperability between systems
  - have no known principle short comings: “everything should be possible”
- key idea: combine best parts of the existing frameworks
  - xml/property files define parameters
  - java driver classes use parameters to build detector geometry
  - use **gcj** to compile java to true machine code for C++ frameworks
  - **gcjh** creates C++ header files
  - -> have identical code basis used in Java and C++ frameworks

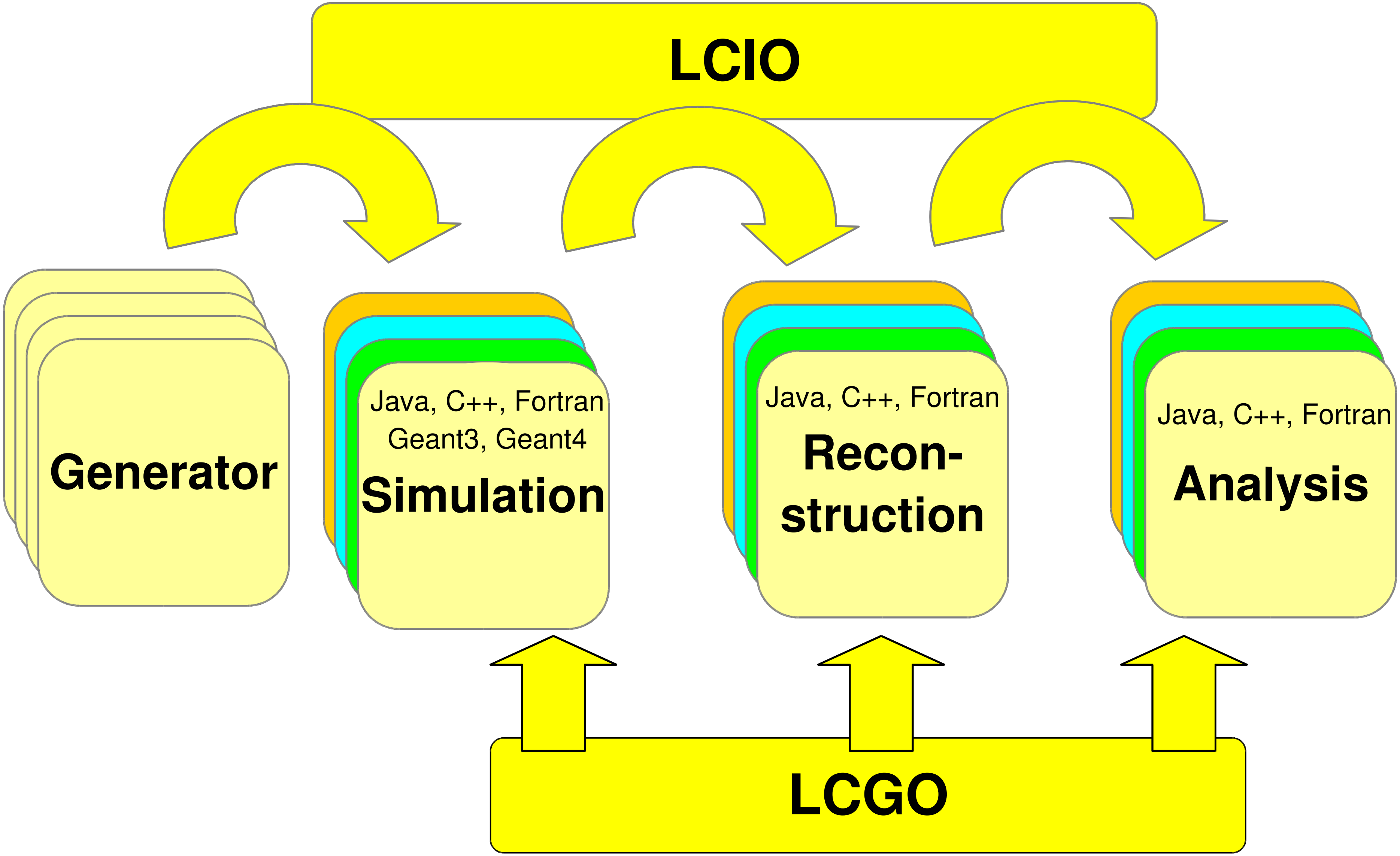
LCGO implementation prelim.



some LCGO planned features

- extended GEAR interface (medium and high level)
  - tracking (and clustering PFA)
    - average material volumes
    - intersection with 'next' volume
    - dE/dx
    - field maps
    - access to volumes
    - extensions of detectors (a la gear)
      - e.g. #layers, thickness, width,..
  - material database
  - field maps
  - properties (sampling fractions)
  - readout properties
    - cellId <-> position
    - cellid range (noise simulation)
    - cell sizes
    - neighbors
- detailed geometry:
  - have geant4 like geometry classes with a one to one correspondence:
    - logical/physical volumes
    - placements
    - materials

ILC interoperable software chain



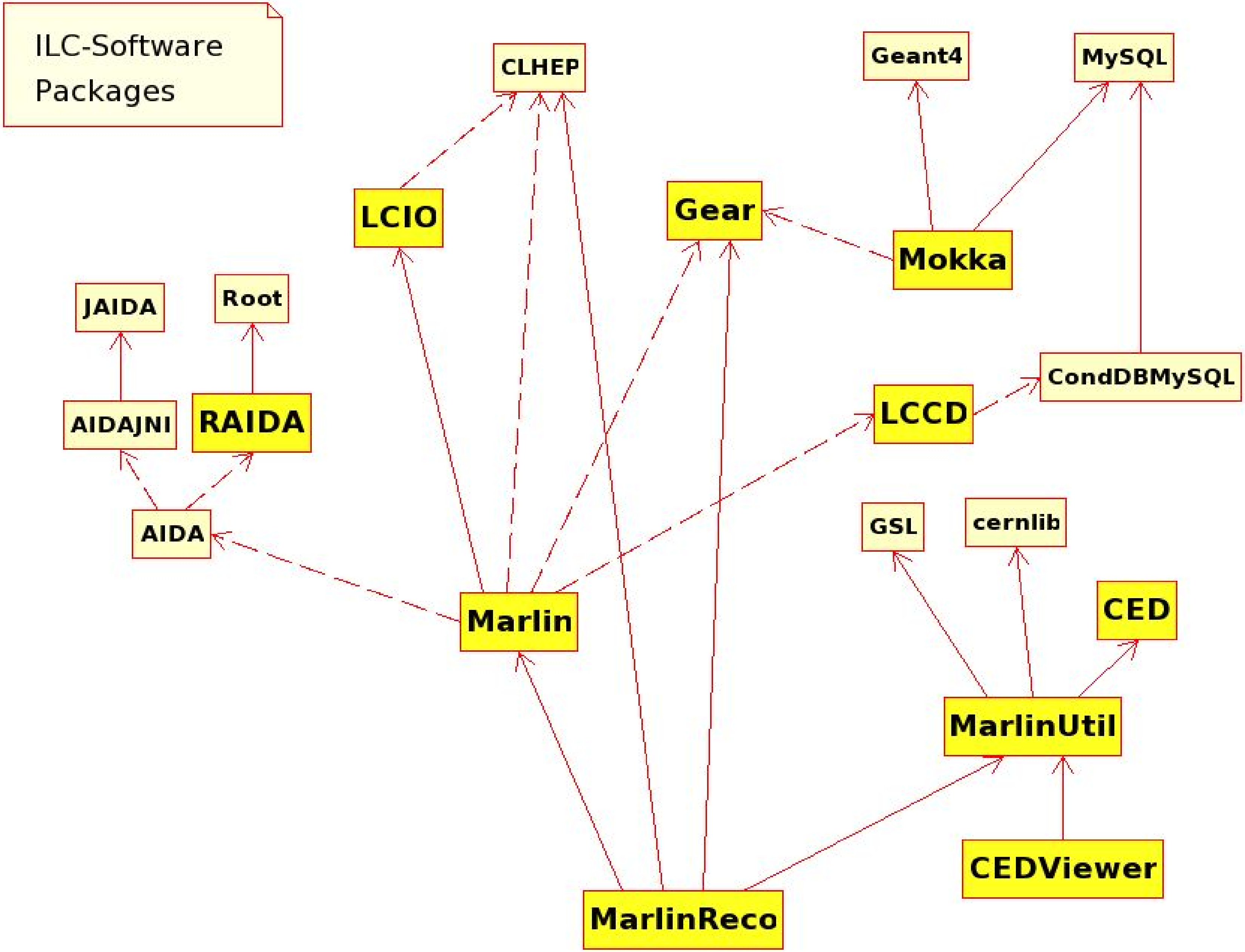
Summary

- the ILC/LDC software framework with LCIO, Marlin, LCCD, GEAR is under active development
- growing user community
  - Calice +TPC testbeams, PandoraPFA, LCFI-ZVtop,...
- new common geometry toolkit LCGO under development:
  - increase interoperability with other systems
  - simplify modification of geometry for concept studies
  - allow comparisons across frameworks and concepts !

- please consider using common software tools
- provide feed back (bugs, new features, requirements)
- contribute and provide your code to the community

Backup Slides

software package dependencies



Marlin core features

- fully configurable through steering files:
  - program flow
  - input parameters (processor based and global)
- self-documenting:
  - ./bin/Marlin -x

    prints example steering file with

    all available processors with their parameters and example/default values
- AIDA interface for histogramming
  - easy creation of histograms through absract interface
  - AIDAJNI/JAIDA, RAIDA (root based), ...
- configurable output
  - drop collections by name/type
- simple examples
  - user processor template, GNUmakefile,...
- easily extensible
  - makefiles 'automatically' include user packages with processors