

Status of the ILD SW framework iLCSoft

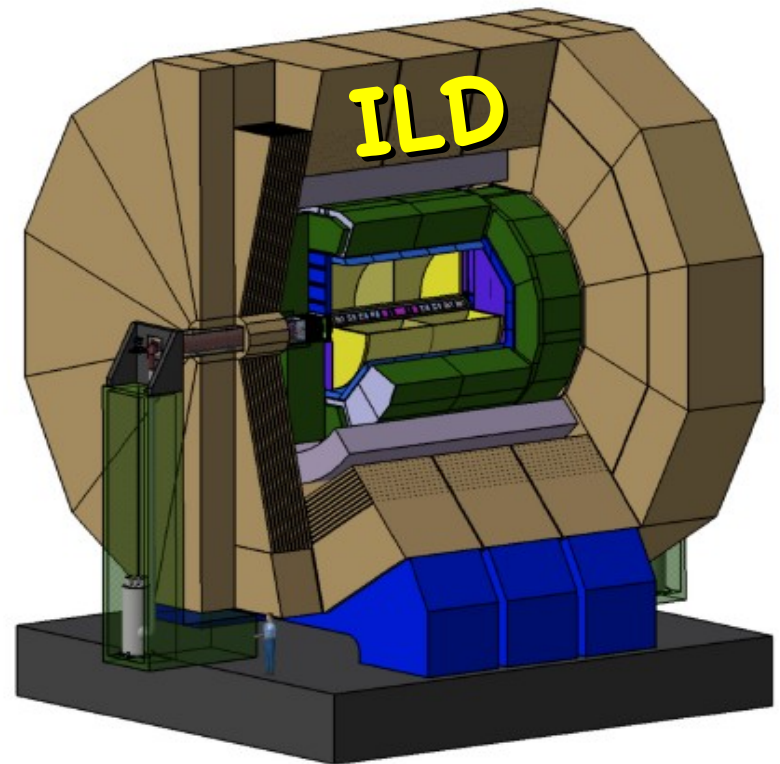
Frank Gaede, DESY

ALCPG 2011

Eugene, March. 19-23, 2011

Outline

- Introduction
- Overview ILD software chain
- recent developments (since LCWS2010)
 - core tools
 - simulation (ILD_01)
 - reconstruction tools
- AIDA WP2
- Summary & Outlook



iLCSoft development timeline

- iLCSoft used by CLIC and ILD
- two different timelines:
 - CLIC CDR – mid 2011
 - ILD DBD – end 2012
- CLIC is using what exists now
- ILD has now time to further develop and improve the software to fulfill the needs for the DBD (e.g. improved realism in detector simulation)

5 month	Analysis and Writing	13 month
t0 - 5m	Monte Carlo production finished	
5 month	Grid Production	
t0 -10m	start Monte Carlo production	
3 month	Test, Debug and release ILDsoft	
t0-13m	freeze ILDsoft development	~20 month
>1 month	implement baseline in simulation	
t0-x	ILD baseline defined	
	evaluate technology options develop tracking package develop geometry LCIOv2	
	improve simulation realism improve reconstruction study machine backgrounds	

ILD Core Software Tools

<http://ilcsoft.desy.de>

- **Mokka** (LLR)

- geant4 simulation application

- **LCIO** (DESY/SLAC)

- international standard for persistency format / event data model

- **Marlin**

- core application framework for reconstruction & data analysis

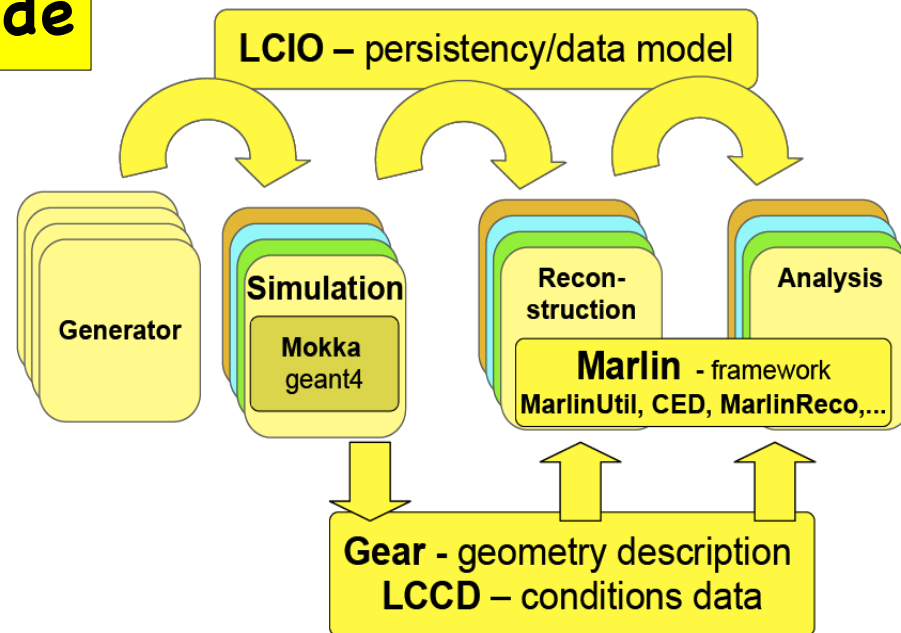
- **GEAR** geometry package f. reconstruction

- **LCCD**

- conditions
- data toolkit (DB)

- **CED**

- 3d event display



- complete framework used in Monte Carlo & 'real experiments':

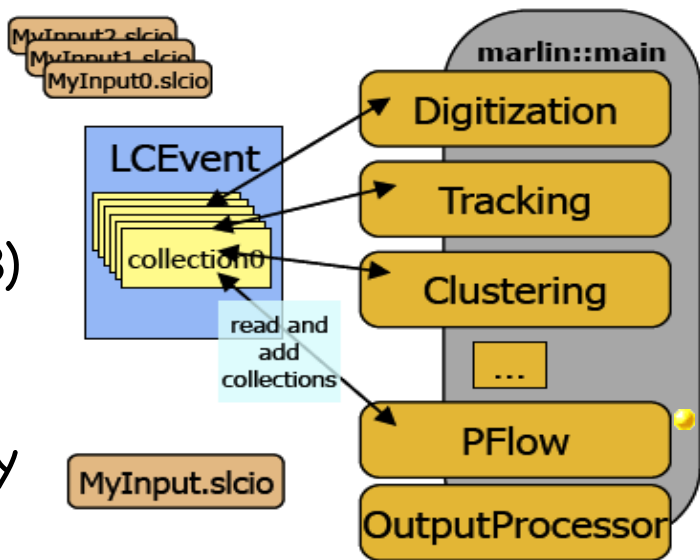
- **ILD detector concept** studies

- **Calice** calo testbeam

- **LC-TPC** testbeam

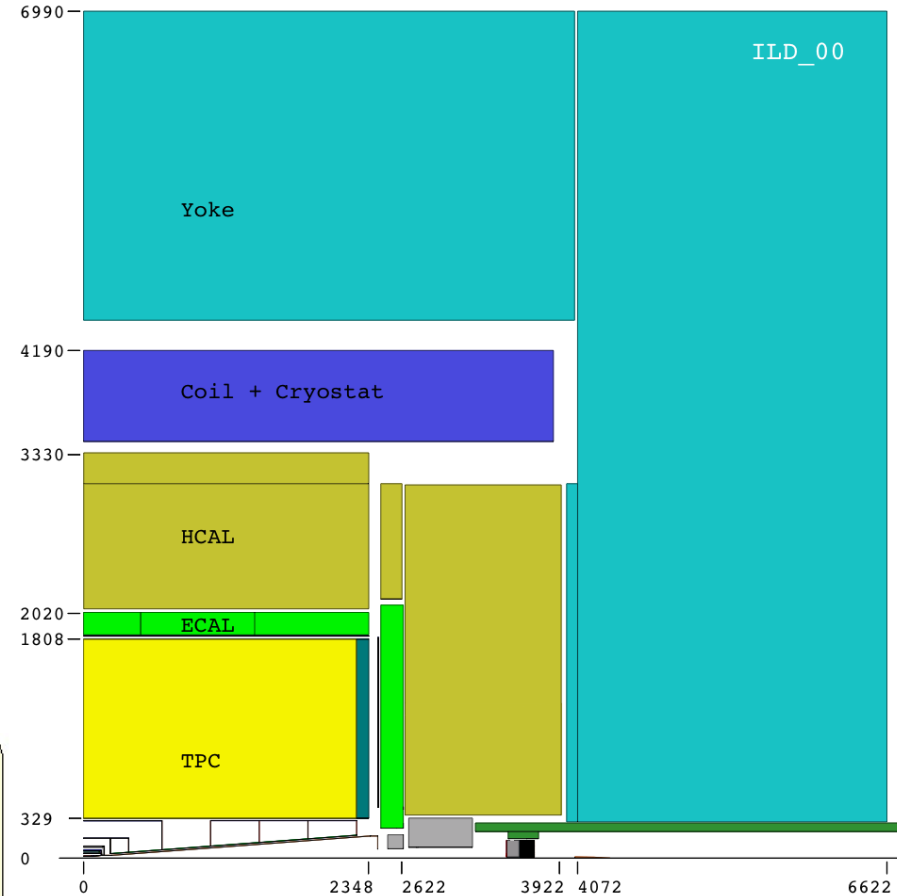
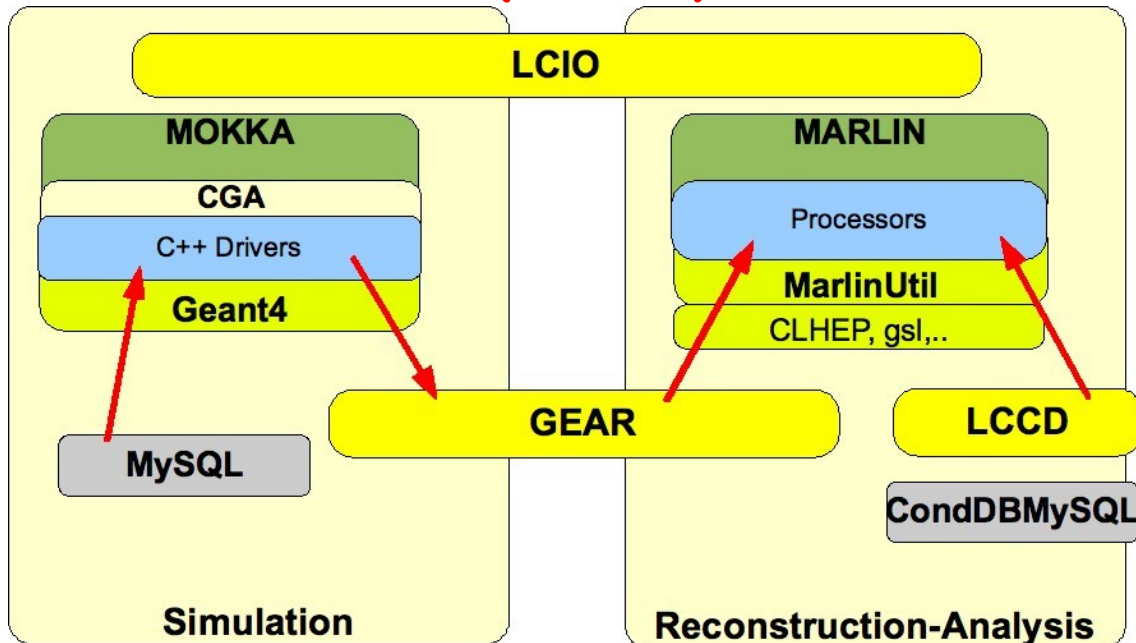
- EUDET - **Pixel Telescope**

• **synergies between testbeam and global detector optimization**



Mokka Simulation ILD

- defined 'ILD simulation reference model' for LOI mass production
- engineering level of detail for **most** subdetectors:
 - support structures
 - cracks
- modified to ILD_CLIC detector model to be used for CLIC CDR !
- **goal: further improve realism of ILD model for DBD (ILD_01)**



Mokka writes out GEAR xml files with complete geometry and material parameters that are need for reconstruction and analysis

Digitization & Reconstruction in Marlin

• VXD, SIT, FTD, SET, ETD

- smearing of 3D space points according to detector resolutions as established by R&D groups

• TPC hits

- smearing of 3D space points - taking into account drift distance, polar and azimuthal angle of track
- parameterization from TPC R&D groups

• ECal, HCal, LCal, Bcal, LHCAL, Muon Calo hits

- calibration (single particle resolution)

• Tracking

- standalone tracking in Silicon detectors and TPC - **MarlinReco-FullLDCTracking**
- Kalman filtering: **wrapped f77 code from LEP**
- **new development started ...**

• Particle Flow Algorithm

- **MarlinPandora/PandoraPFANew**

• JetFinder

- Durham jet finder (run for 2-6 jets)

• Flavour Tagging

- **LCFIVertex** package: ZVTop, ZVRes + Neural Network Fl.Tag

• DST Maker

- ReconstructedParticles, Jets, Tracks and Clusters (25k/evt)

ILD software builds and installation

- **ilcinstall** tool: python scripts to download, build and install all ILD and external packages – incl. test beam
 - complete iLCSoft – provided **geant4, root and mysql** are installed
- used for
 - **iLCSoft releases** (ilcinstall release tag == ilcsoft release)
 - **reference installations** in afs
 - **grid installations** (all WLCG sites supporting VO ILC)
 - **binary tar-balls** (SL4/5)
- started to have more frequent 'developers' releases

afs reference installations

- provide reference installations in afs for usage from **anywhere** on ScientificLinux and compatible platforms:

`/afs/desy.de/project/ilcsoft/sw/_OS_/v01-10-01`

```
_OS_ : i386_gcc34_sl4      # i386 CPU, 32 bit, gcc3.4, SL4 and compatible
       i386_gcc41_sl5      # i386 CPU, 32 bit, gcc4.1, SL5 and compatible
       x86_64_gcc41_sl5    # i686 CPU, 64 bit, gcc4.1, SL5 and compatible
```

- you can directly run from these installations
- you can link your own libraries against these
- plan to have other OSs in the future (as needed)

Note: older releases (<v01-09) at
`/afs/desy.de/ilcsoft/`

ILCTest

S.Aplin, J.Engels

- generic test system for iLCSoft:

- unit tests
- integration tests
- physics test

- added some unit tests to most packages (run in Nightly Builds)

- result browsable on dashboard
- can be added to any iLCSoft package

```
// first line in your c++ source file
static ILCTest ilctest = ILCTest( "hello_world" );
...
ilctest.log( "hello world test" ); // a log message
...
If( x != 42 ){ ilctest.error("wrong answer!!") ; }
...
cout << ilctest.last_test_status() << endl; // prints "FAILED"
...
If( r > 3 ){ ilctest.fatal_error("this is a fatal error. program will quit now!") ; }
```

Frank Gaede, ALCPG 2011, Eugene, March 18-23, 2011

My CDash | All Dashboards | Log Out
Wednesday, September 15 2010 11:25:19 CEST

LCIO Dashboard

DASHBOARD CALENDAR PREVIOUS CURRENT PROJECT ADMINISTRATION

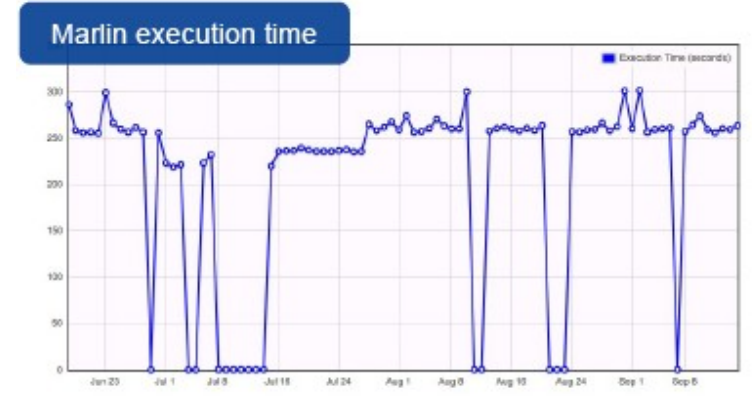
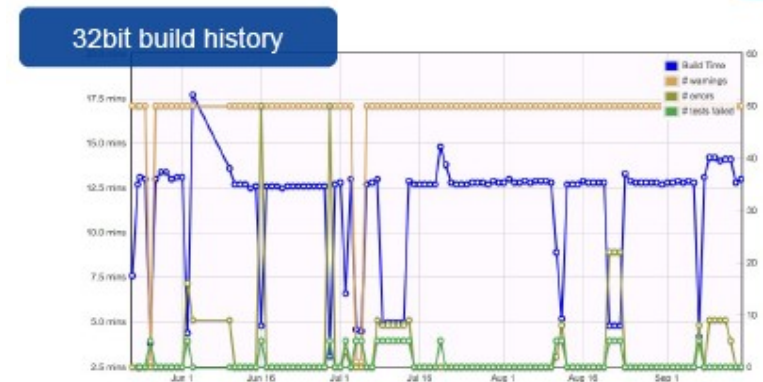
No file changed as of Wednesday, September 15 2010 00:00:00 CEST [Help](#)

[Show Filters]

Nightly

Site	Build Name	Update		Configure			Build			Test				Build Time
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min	
grid-llc-pa0	linux-gcc-debug			0	0	0	0	12	0.1					2010-09-15T02:01:57 CEST
grid-llc-pa0	linux-gcc-debug-x64			0	0	0	0	12	0.1					2010-09-15T04:01:42 CEST
grid-llc-pa0	linux-gcc-default			0	0	0	0	12	0.2					2010-09-15T02:02:08 CEST
grid-llc-pa0	linux-gcc-default-tests	0	0.1	0	0	0	0	13	0.3	0	0	21	0.3	2010-09-15T02:01:11 CEST
grid-llc-pa0	linux-gcc-default-tests-x64	0	0.1	0	0	0	0	13	0.2	0	0	21	0.2	2010-09-15T04:01:08 CEST
grid-llc-pa0	linux-gcc-default-x64			0	0	0	0	12	0.2					2010-09-15T04:01:54 CEST
Totals	6 Builds	0	0.2	0	0	0	0	74	1.1	0	0	42	0.5	

No Continuous Builds



current iLCSoft release: v01-10-01

CED	v01-01-01
CEDViewer	v01-01-01
CLHEP	2.0.4.5
CMakeModules	v01-10-01
CondDBMySQL	ILC-0-9-2
Druid	1.8
Eutelescope	v00-04-04
LCFIVertex	v00-04
LCFI_MokkaBasedNets	v00-01
Marlin	v00-13-01
MarlinPandora	v00-02
MarlinReco	v00-19-01
MarlinTPC	v00-06
MarlinUtil	v01-01-pre
Mokka	mokka-07-06
MokkaDBConfig	v02-01

Overlay	v00-08
PandoraPFA	v03-02-02
PandoraPFANew	v00-04
QT	4.2.2
RAIDA	v01-05
SiliconDigi	v00-04-02
StandardConfig	v02-01
cernlib	2006
dcap	1.9.5-5
gear	v00-16
gsl	1.8
lccd	v01-01-01
lcio	v01-51-02
mysql	5.0.45
root	5.26.00b
KalTest	v01-00
KalDet	v01-00

- many packages changed (wrt v01-09)
- **some new added**

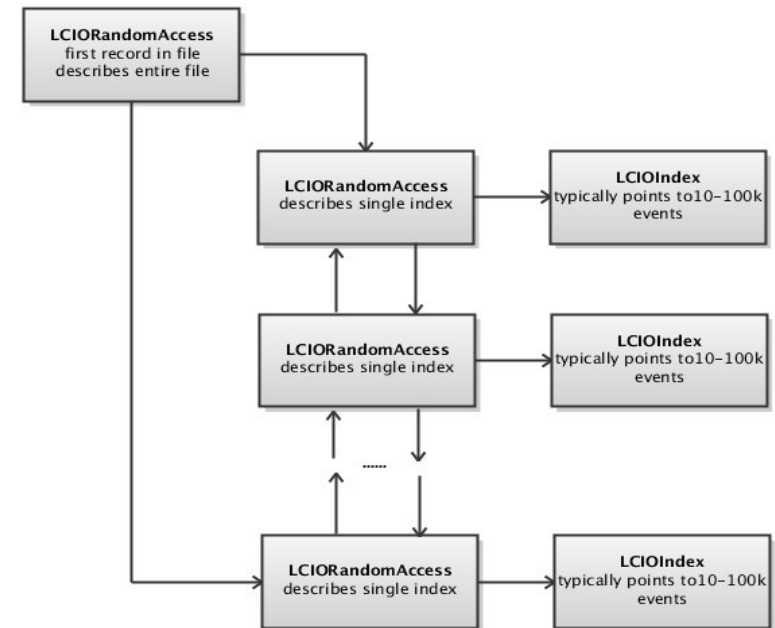
currently: v01-11 under preparation 10

towards LCIOv2 - v01-51-03

- LCIO provides a **rather complete event data model** and has been used successfully in SID and ILD LOI mass production and in various R&D test beam programs
- LCIOv2 needs to be backward compatible and should provide some new features

- **direct access to events -> DONE**
- **partial reading of events**
- **splitting of events over files**
- **(storing of arbitrary user classes)**
- **simplify using LCIO with ROOT -> DONE**
 - (ROOT macros, TTreeView, I/O (?), ...)
- **improving the event data model**
 - (1d,2d hits, tracks/trajectories)
 - **-> this meeting (see: talk T. Johnson)**

- new ostream operators <<(...) in C++
- `cout << ((MCParticle*) c->getElementAt(i)) << endl ;`



direct Access:

- record written at close()
- can append to files
- **can add to existing OLD files**
(if opened in write mode) 11

a ROOT dictionary for LCIO

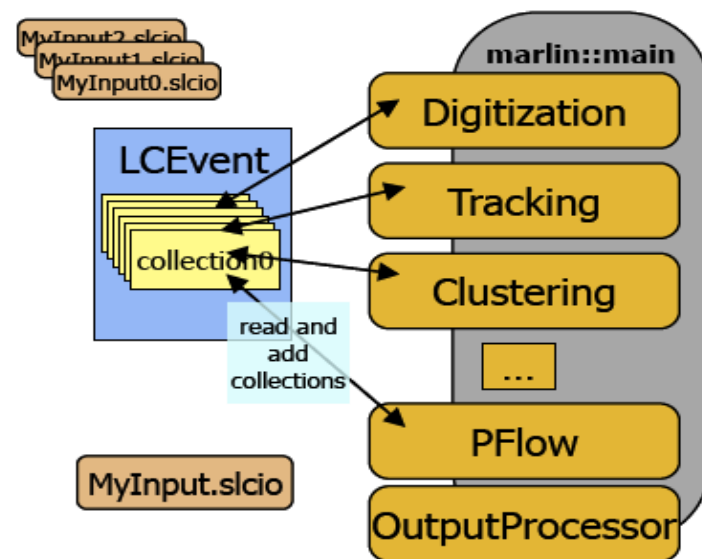
- LCIO now comes with a ROOT dictionary for all LCIO classes (optional) - with this one can:
 - use LCIO classes in ROOT macros (already in v01-12-01)
 - write simple ROOT trees, e.g. `std::vector<MCParticleImpl*>`
 - use TTreeDraw for quick interactive analysis of LCObjects:

```
//---gamma conversions:  
TCut isPhoton("MCParticlesSkimmed.getPDG()==22" );  
LCIO->Draw("MCParticlesSkimmed._endpoint[][0]:  
          MCParticlesSkimmed._endpoint[][1]",isPhoton ) ;
```

- write complete LCIO events in one ROOT branch
- see: [\\$LCIO/examples/cpp/rootDict/README](#) for details & help
- -> we are interested in feedback from the users if this provides already the requested features

new Marlin releases

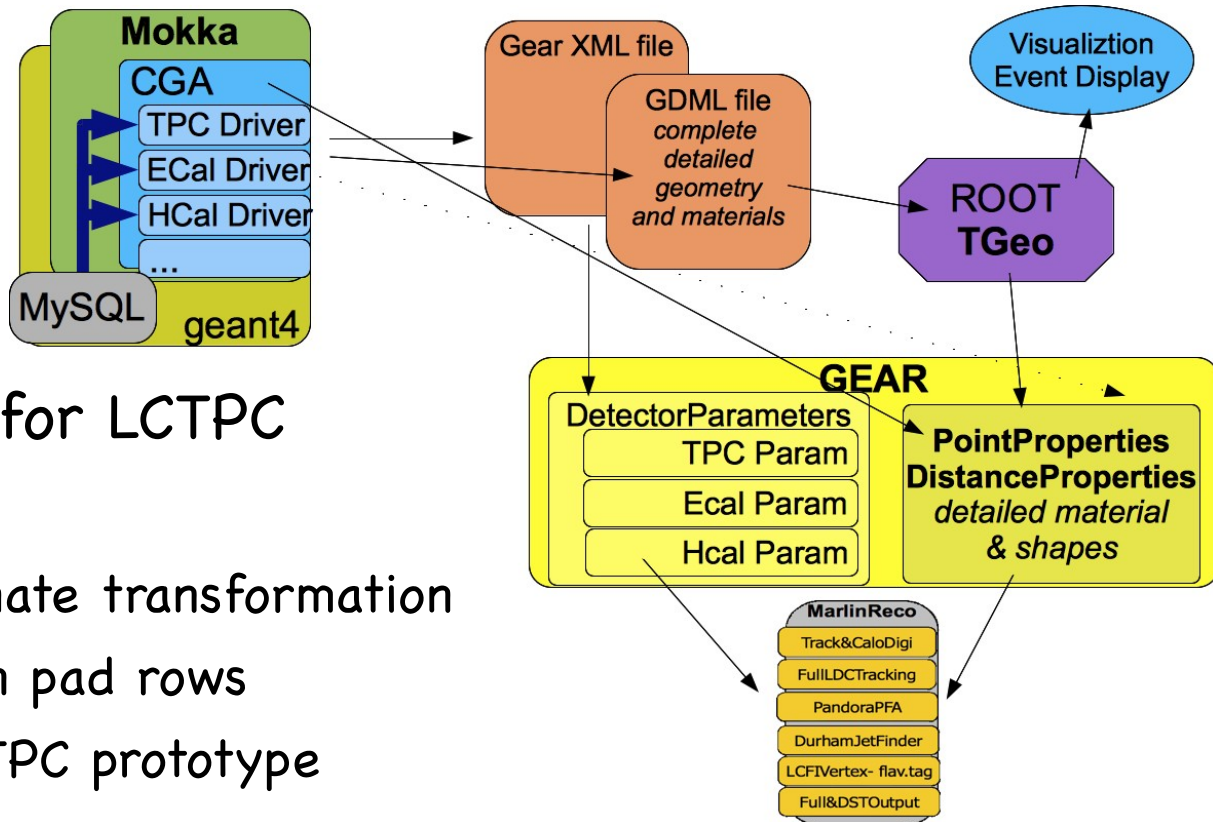
- new features in v00-12/13
 - processors can have **local Verbosity** parameter
 - can turn off messages from other processors for debugging
 - **exit if processor specified in steering file not found**
 - + minor fixes & improvements



- next release (v01-00)
- **new RandomSeeder:** (S.Aplin)
 - provide random seed per event and processors
 - -> reproducible event processing even with SkipEvents (usefull for debugging)
- **command line options (a la JSF)**
 - under development (J.Engels)

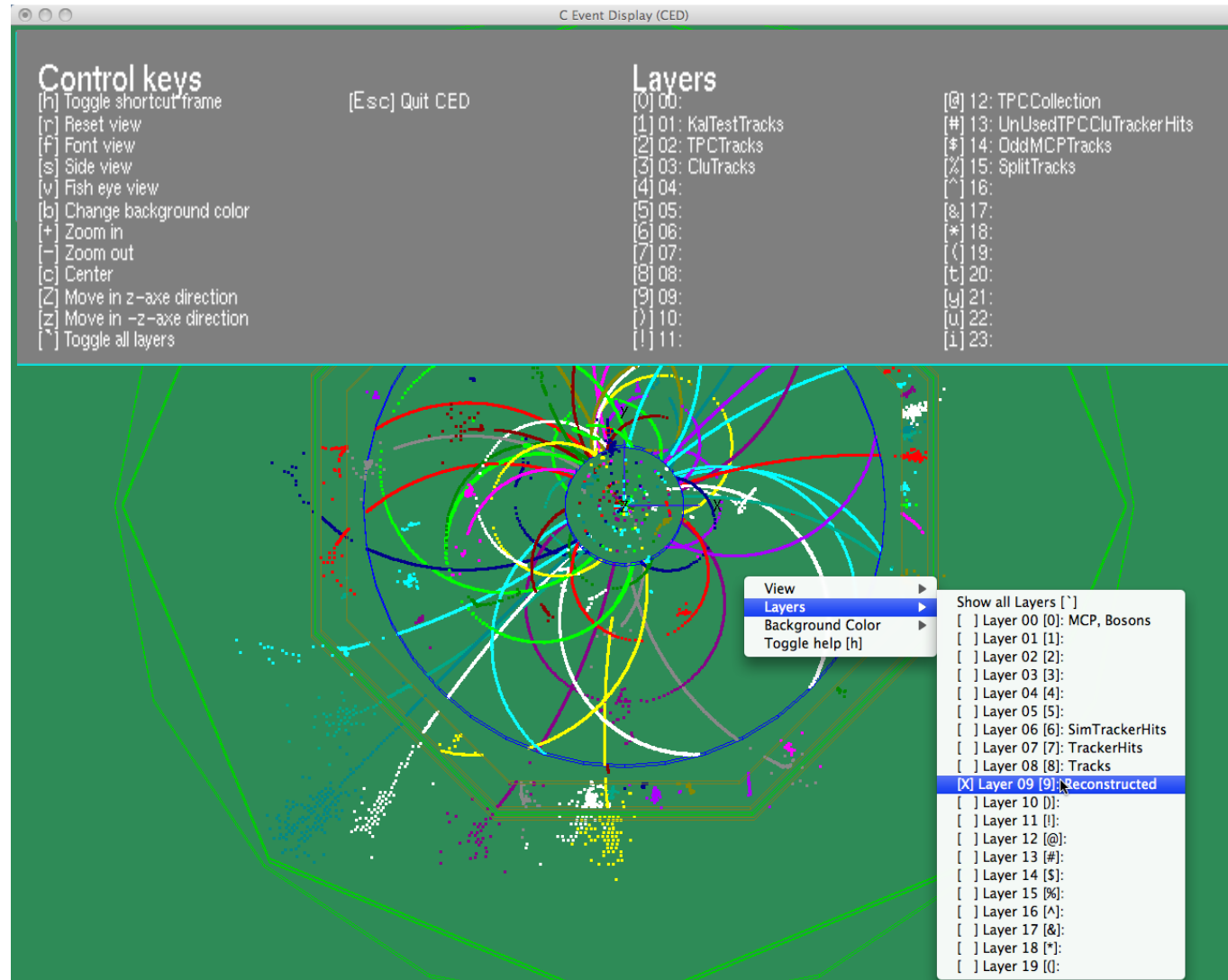
GEAR release - v00-15

- new in v00-15:
- made pure C++ (no Java)
- improved TPCParameters for LCTPC (M.Killenberg)
 - improved global-local coordinate transformation
 - introduced circle segments in pad rows
 - -> needed for describing LCTPC prototype
- **Point/DistanceProperties implemented with TGeo** (A.Muennich)
 - uses GDML interface to geant4
 - introduced material map in memory (performance)
- Mokka-CGA (geant4) implementation of Point/DistanceProperties released in Mokka



improved CED event display - I

- added help menu
 - toggled with 'h'
 - shows all keys
 - shows all 'collections'
- added mouse menu
 - toggle single visualization layers
 - choose bg colors
 - views
 - zoom
- commands to add layer description and picking also for user code
- new python script to start CED & CEDViewer in on go: [ced2go.py](#)



new Mokka release - ILD_01_pre01

- major rewrite of some sub detector drivers :

- SIT, SET, ETD - FTD - Muon

- **increased level of detail and realism (incl. services)**

- TPC

- added endcap services (cooling)

- new ECal driver:

- mixing of Scintillator and Si layers

- improved aHcal driver:

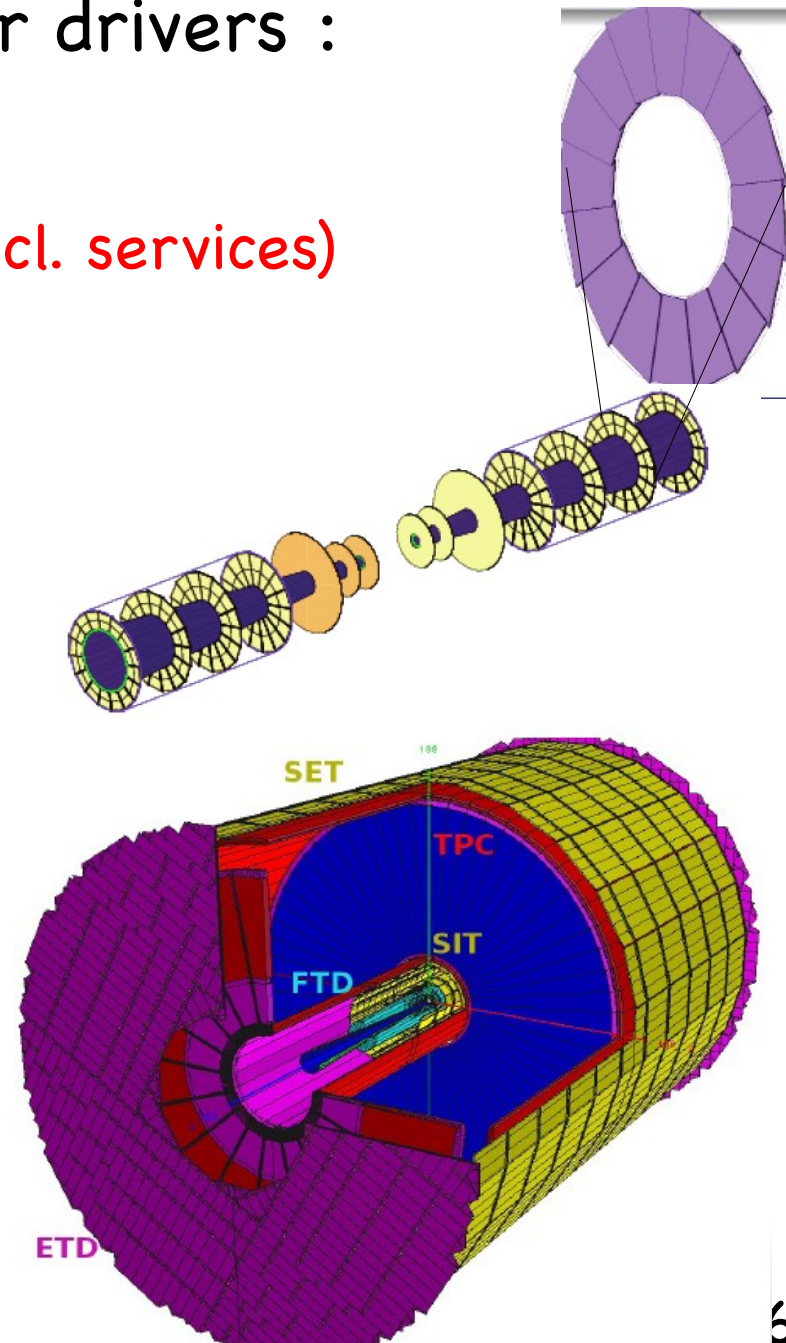
- included electronics & services

- **overall services for TPC, Ecal, Hcal**

work of many people:

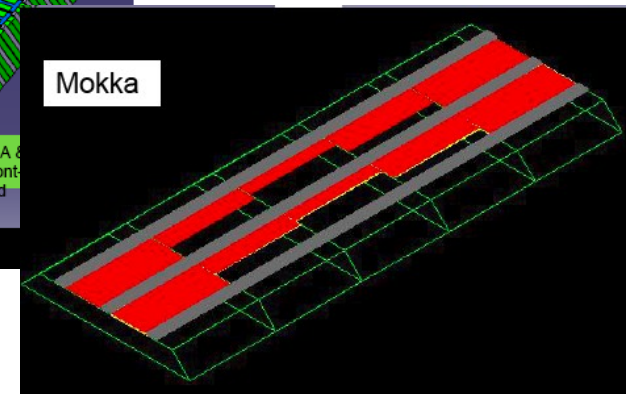
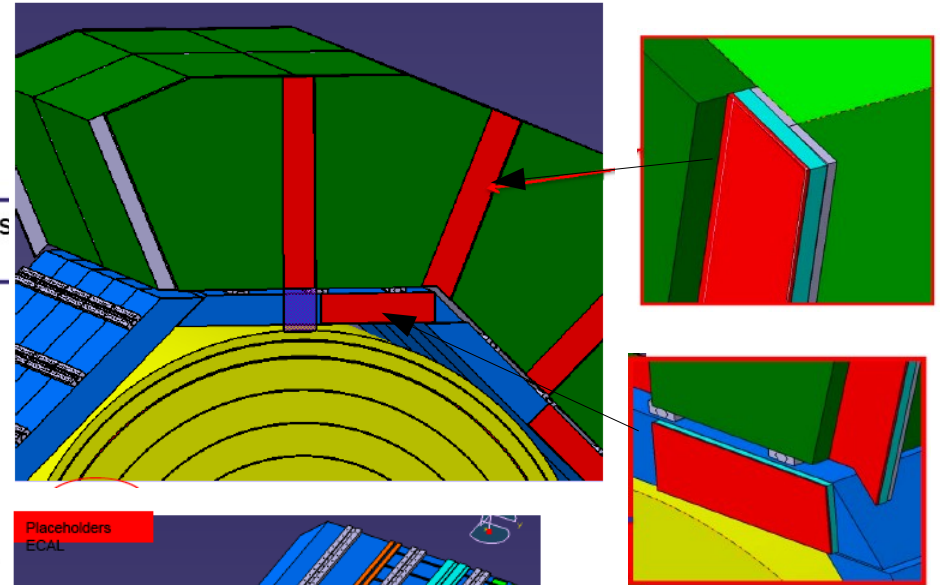
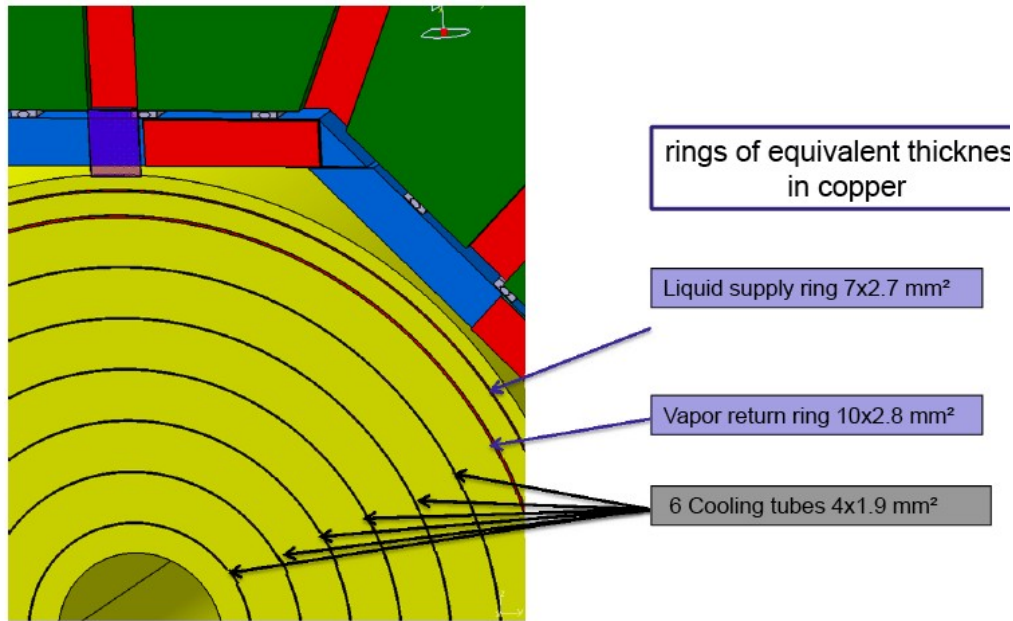
A.Charpy, J.Duarte, A.Saveliev, G.Musat,

A.Lucaci, P.Mora de Freitas,....



new Mokka release – towards ILD_01

- added cabling and services for TPC, ECal & Hcal (C.Clerc, G.Musat)
- still missing: inner detector services (to be defined by R&D groups)

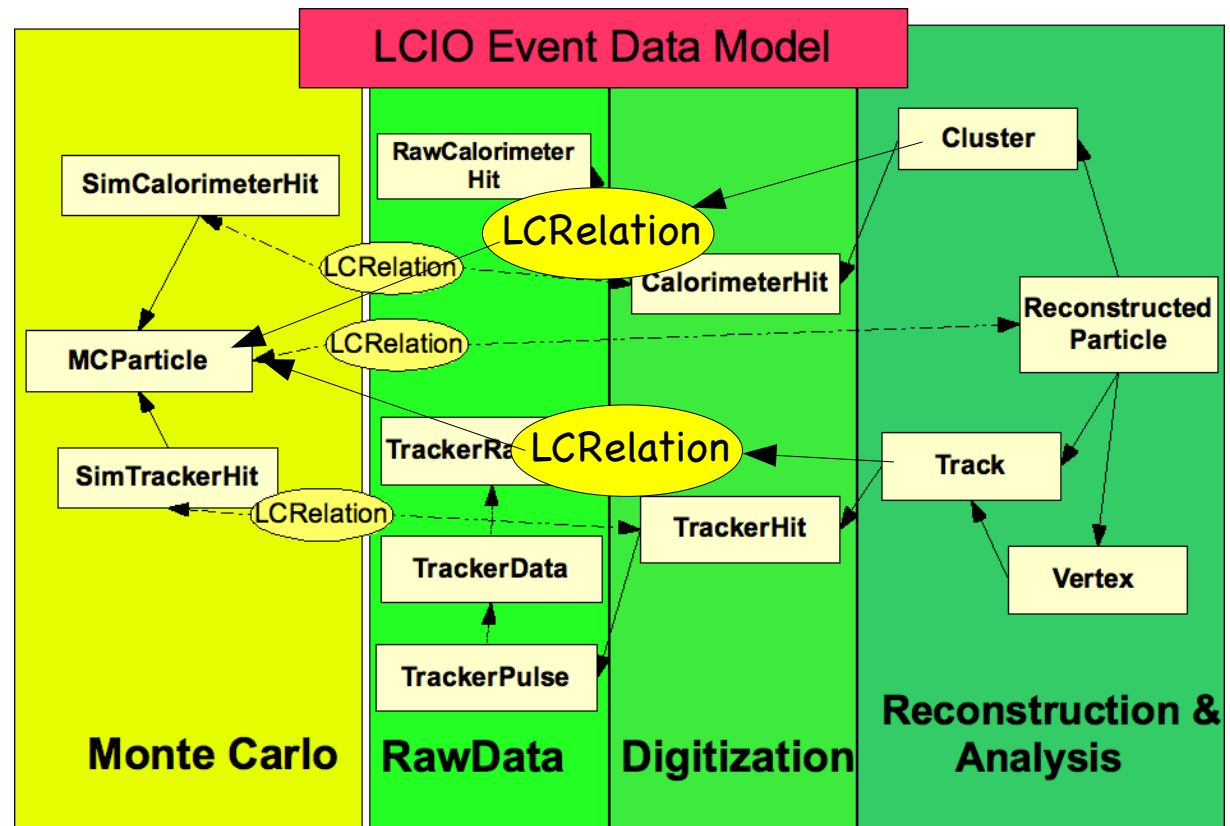


big step forward in increasing realism of ILD detector simulation !

MarlinReco - v00-19-01

- new package **KinkFinder** (M.Thomson, J.Marshall)
- new package **BCalTagEfficiency** (J.List, M.Berggren)
- new package **FPCDDigi** (D. Kamai)
- + bug fixes and improvements by many people ...

- **RecoMCTruthLinker**
 - added additional relations between **MCParticle** and **Tracks** and **Clusters** - to be used for **DST** (M.Berggren)
 - => more complete information for efficiency studies !



new packages in iLCSoft - v01-10

- **KalTest**

- Kalman Fitting library (Keisuke Fuji et al)
 - migrated code base to SVN
 - added cmake build scripts

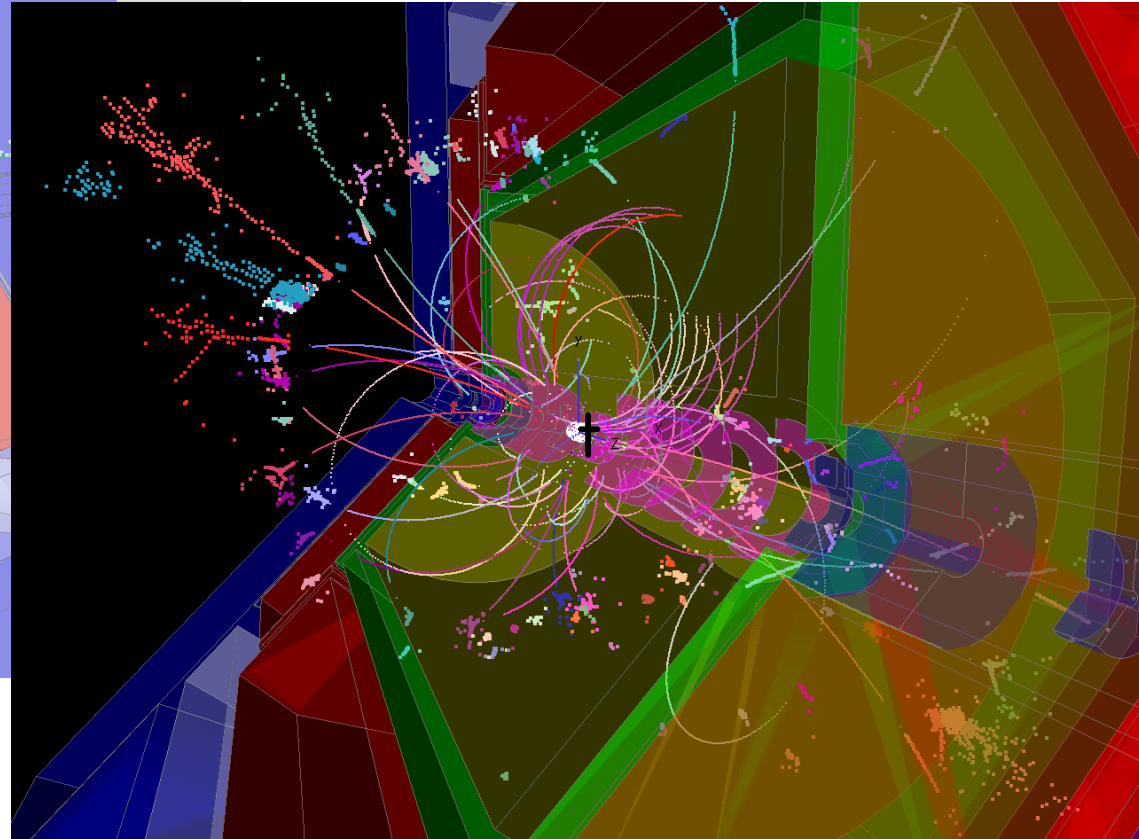
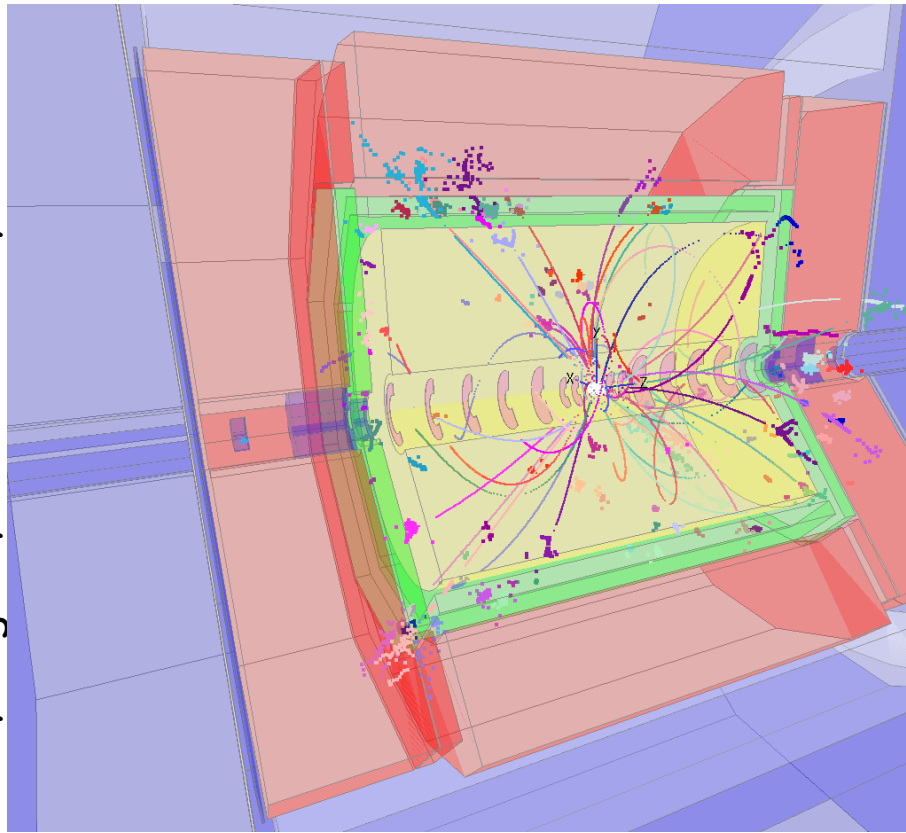
- **KalDet**

- detector description (geometry and material) for KalTest
 - currently writing the geometry build up from GEAR
- both packages will be used by LCTPC (MarlinTPC) and ILD
- ILD started development of new TPC tracking
- (-> see talk S.Aplin)
- -> try to share as much common code as possible, i.e. is reasonable given the slightly different requirements for testbeam and global detector optimization

towards v01-11

- currently v01-11 is under preparation (v01-11-pre03)
- mainly targeted at providing a software release for the **CLIC CDR Monte Carlo mass production** - including:
 - OverlayTiming (CLIC background studies)
 - improved tracking
 - split tracks, forward tracks, increased common blocks, z-slices in TPC
 - special digitizers (timing information)
 - steering files for ILD-clic reconstruction
 - background filters (gamma gamma -> hadrons)
- -> also some improvements to core/build tools ...

improved CED event display - II



- added **client server functionality**
 - run CED on different machines
 - (t.b.r. in v01-11)
 - add 3d volumes/shapes w. clipping for nice pictures
-> under development

new package ILCUTIL

- (meta) package with utility packages:
- **ILCSOFT_CMAKE_MODULES**
 - (Previously known as CMakeModules)
 - cmake modules and utility macros.
 - NB: almost all iLCSoft packages depend on this !
- **ILCTEST**
 - C++ utility headers
 - cmake macros
 - for unit and integration tests of iLCSoft packages
- **streamlog**
 - logging classes for C++ applications
 - used to live in Marlin -> can now be used in other packages
- -> can be extended with other useful general purpose utilities

simplified use of CMake in iLCSoft

- iLCSoft uses CMake as build tool
- now greatly improved:
 - deprecated: BuildSetup.cmake, \$PKG_HOME, -BUILD_WITH, LoadPackage.cmake and CheckDeps.cmake
- simply use: FindPackage()

Typical FIND_PACKAGE usage

- FIND_PACKAGE(Marlin REQUIRED)
- INCLUDE_DIRECTORIES(\${Marlin_INCLUDE_DIRS})
- LINK_LIBRARIES(\${Marlin_LIBRARIES})

-> also includes needed packages
LCIO, Gear, RAIDA, etc !!

Version checking

- FIND_PACKAGE(ROOT 5.28 REQUIRED)
- FIND_PACKAGE(LCCD 1.2 EXACT)

Using COMPONENTS

- FIND_PACKAGE(ROOT 5.28 REQUIRED COMPONENTS Gdml Geom XMLIO)
- LINK_LIBRARIES(\${ROOT_LIBRARIES})
- LINK_LIBRARIES(\${ROOT_COMPONENT_LIBRARIES})
- LINK_LIBRARIES(\${ROOT_GDML_LIBRARY})

for details see: [http://ilcagenda.linearcollider.org/getFile.py/access?contribId=1
&resId=0&materialId=slides&confId=4950](http://ilcagenda.linearcollider.org/getFile.py/access?contribId=1&resId=0&materialId=slides&confId=4950)

AIDA WP2 – Common Software

develop core software tools that are useful for the HEP community at large and in particular for the next big planned projects: sLHC and **Linear Collider (ILC/CLIC)**

Task 2.2: Geometry toolkit for HEP

- Allow the description of complex geometrical shapes, materials and sensitive detectors
- Provide interfaces to full simulation programs (Geant4), fast simulations, visualization tools and reconstruction algorithms
- Allow for the misalignment of detector components
- Provide an interface to calibration constants and conditions data

- start: Feb 2011
- for 4 years

Task 2.3: Reconstruction toolkit for HEP

- Tracking toolkit based on best practice tracking and pattern recognition algorithms
- Provide alignment tools
- Allow for pile up of hadronic events
- Calorimeter reconstruction toolkit for highly granular calorimeters based on Particle Flow algorithms

general strategy:

- **integrate as much as possible with existing software framework(s) and international activities outside of the AIDA project**
- collaborate with software activities in other AIDA work packages, e.g. the alignment of silicon sensors

Summary & Outlook

- since LOI we started to work on improving the software for the DBD
- many developments in core software tools have been done in 2010
 - first version of ILD_01 model with improved realism on the way
 - improved LCIO (first step towards v2)
 - new GEAR implementation based on TGeo
 - many improvements and fixes in MarlinReco
 - improved CED
 - first version of new test system
- released as **v01-10-01** (- v01-011 soon)
- Next Steps:
 - continue to improve software tools - focus on
 - development of
 - new C++ tracking software
 - getting ILD_01 to work (digitization/reconstruction)