

Common Software

EUDET task ANALYS

Frank Gaede, DESY
EUDET Annual Meeting
DESY, October 1st, 2010

Outline

- Overview
 - task ANALYS
 - iLCSoft framework
- recent developments in core software
 - LCCD, test system
- JRA software frameworks
 - EU Telescope
 - MarlinTPC
 - Calice Soft
- AIDA WP2
- Outlook

EUDET task ANALYS

- ANALYS: development of a **common data analysis and simulation infrastructure**
 - development of a **software framework** for simulation, analysis and comparison of test beam experiments
 - -> have "version 1.0" of framework after 18 months
 - **embedded into existing GRID infrastructure**
- **strategy**
 - the test beam software effort is tightly integrated with the **overall common ILC (ILD & CLIC) software effort !**
 - **benefit from synergies where possible**
 - same for grid: integrate with common ILC grid activities

EUDET Funding for ANALYS

- originally only 8ppm in proposal for ANALYS (core software)
 - → would not have been enough for the tasks
- throughout the project some leftover funds re-allocated to software
 - in particular 30 kEUR last year from TA
 - → very helpful in keeping the person power for core software development up

EUDET & 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

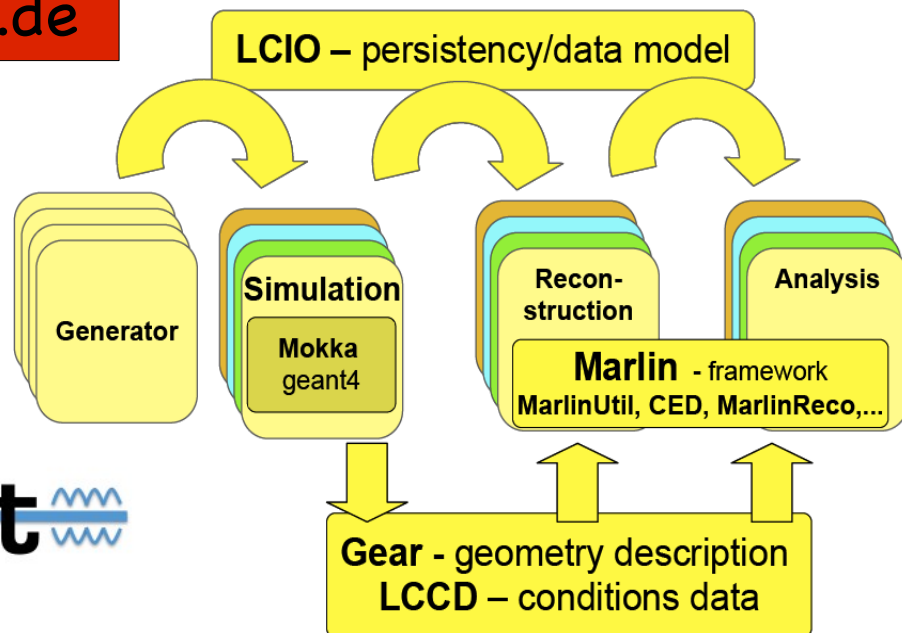
- **GEAR** geometry package f. reconstruction

- **LCCD**

- conditions
- data toolkit (DB)

- **CED**

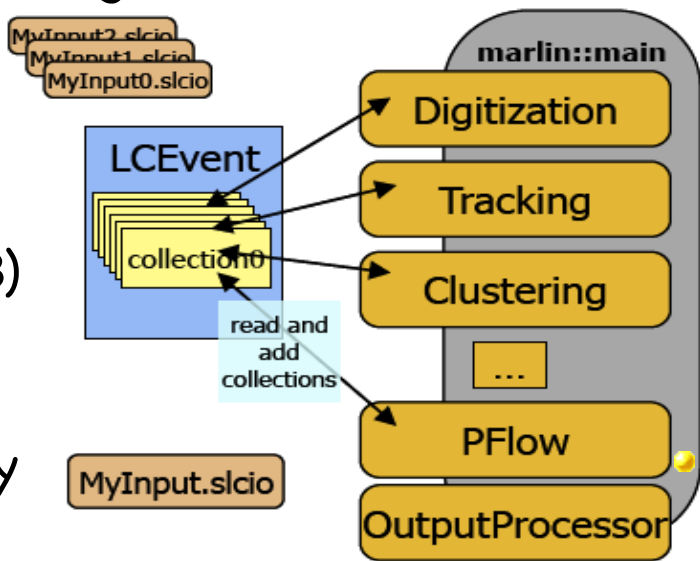
- 3d event display



- complete framework used in Monte Carlo & real experiments:

- **ILD/CLIC** detector studies
- **Calice** calo testbeam
- **LC-TPC** testbeam
- EUDET – **Pixel Telescope**

synergies between testbeam and global detector optimization



Frank Gaede, EUDET Meeting, DESY 01.10.10

some core developments within EUDET

- Marlin
 - MarlinGUI: easy interactive configuration of Marlin jobs
 - logging library, rewind for calibration,
- ilcintstall
 - complete download, build and install of all iLCSoft packages
 - → GRID installations
 - complete iLCSoft releases are installed on all major ILC-Grid sites
 - → reference installations in afs for Scientific Linux
 - use iLCsoft from anywhere in the world w/o installing it
- new Grid production system
 - submit and monitor Grid jobs, provide data catalogue
- LCIO
 - many improvements to event data model for test beams
 - direct access to events
- GEAR
 - extensions to geometry description for test beam setups
- many small feature requests and fixes implemented ...

→ see presentations at previous EUDET meetings

SW-Framework Milestone

Marlin et al - A Software Framework for ILC detector R&D

EUDET-Report-2007-11

- september 2007: v01-00
- 'first' release of fully functional software framework for simulation, reconstruction and analysis of ILC (testbeam) data



F. Gaede*, J. Engels*

December 17, 2007

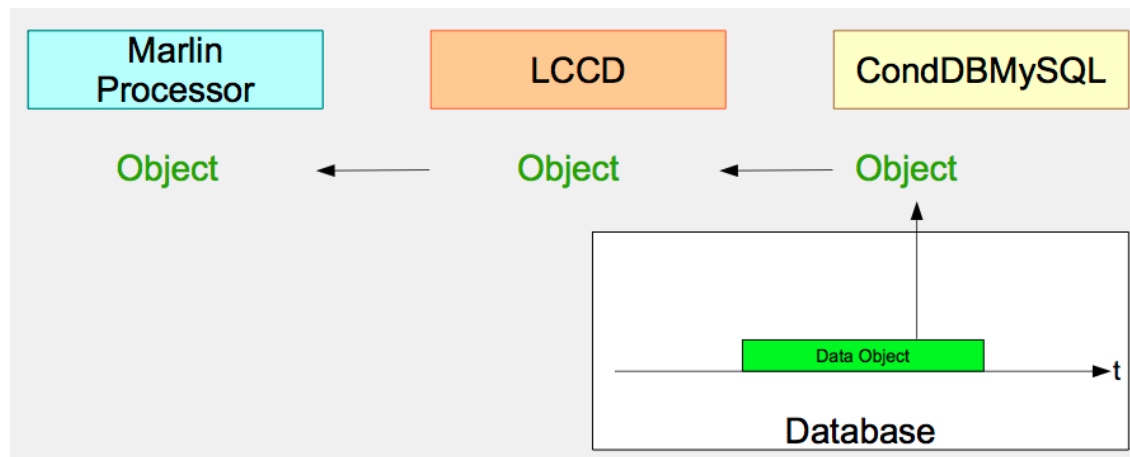
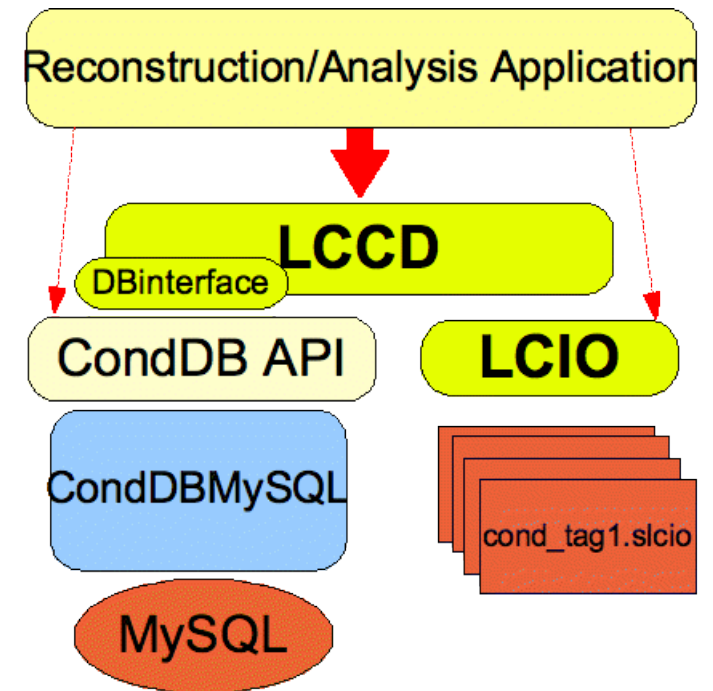
only milestone: "Version 1.0 after 21 month" was reached

=> active development and improvement has continued since

LCCD recent development

S.Aplin

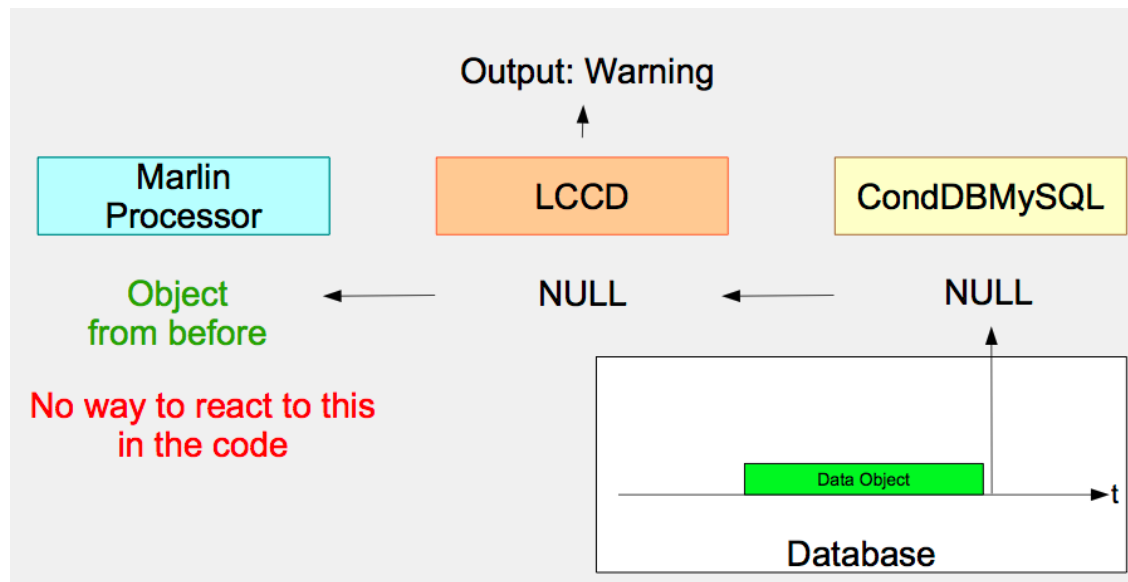
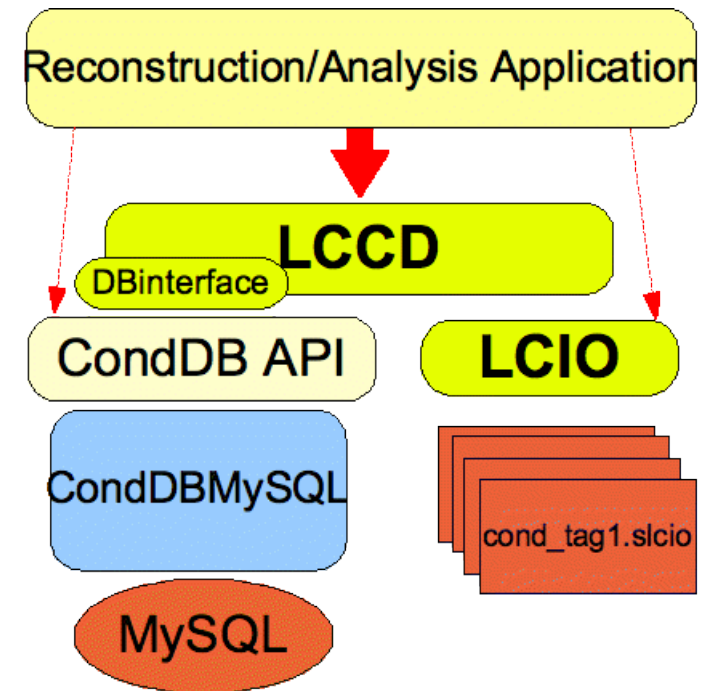
- conditions data toolkit
 - abstract interface to cond. data
 - MySQL-DB, LCIO files, ...
- used by Calice and LC-TPC
- request from the 'real world':
- handle missing conditions data gracefully - originally not foreseen:



LCCD recent development

S.Aplin

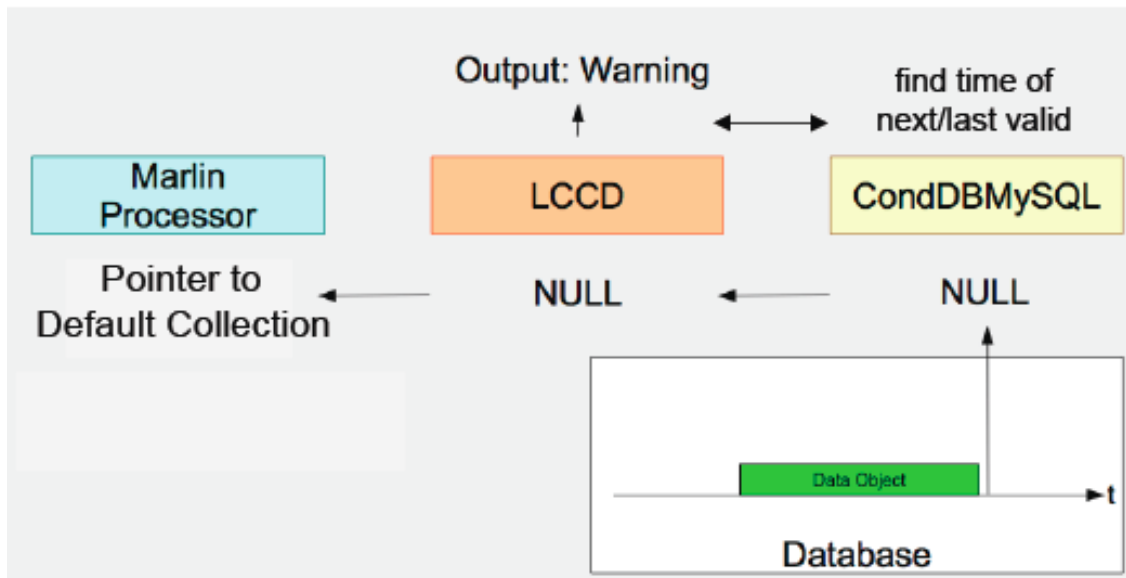
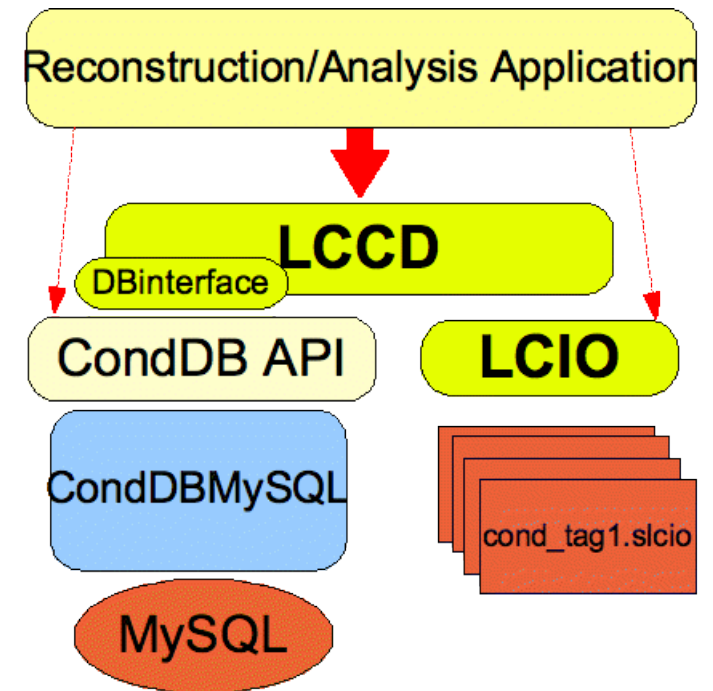
- request from the 'real world':
- handle missing conditions data gracefully - originally not foreseen:
 - either Exception thrown
 - or 'old' conditions data used...



LCCD recent development

S.Aplin

- after iteration with software experts from **Calice & LCTPC**:
- handle missing conditions data gracefully - originally not foreseen:
- return default collection registered by the user
- + information about next valid conditions data



- additional new features:
 - improved folder tagging
 - introduced `lccd::Exceptions` (R.Diener)

new development: ILCTest

S.Aplin, J.Engels

- goal: develop a generic test system for iLCSoft packages with:

- unit tests
- integration tests
- physics test

→ started with creation of dashboard

Name	Status	Time Status	Time	Details
t_c_ana_c2l_rec	Passed	Passed	0.14	Completed
t_c_ana_c_rec	Passed	Passed	0.17	Completed
t_c_ana_c_sim	Passed	Passed	0.09	Completed
t_c_ana_l2c_rec	Passed	Passed	0.13	Completed
t_c_ana_l_rec	Passed	Passed	0.10	Completed
t_c_ana_l_sim	Passed	Passed	0.06	Completed
t_c_rec_c_sim	Passed	Passed	3.78	Completed
t_c_rec_l_sim	Passed	Passed	1.95	Completed
t_c_sim	Passed	Passed	1.99	Completed
t_l_ana_c_rec	Passed	Passed	1.88	Completed
t_l_ana_c_sim	Passed	Passed	1.46	Completed
t_l_ana_l_rec	Passed	Passed	1.41	Completed
t_l_ana_l_sim	Passed	Passed	0.87	Completed
t_l_rec_c_sim	Passed	Passed	2.03	Completed
t_l_rec_l_sim	Passed	Passed	1.63	Completed
t_l_sim	Passed	Passed	0.65	Completed
t_test_calohit	Passed	Passed	0.14	Completed
t_test_example	Passed	Passed	0.00	Completed
t_test_randomaccess	Passed	Passed	0.01	Completed
t_test_trackerhit	Passed	Passed	0.10	Completed
t_test_trackerpulse	Passed	Passed	0.10	Completed

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

[Show Filters]

Nightly

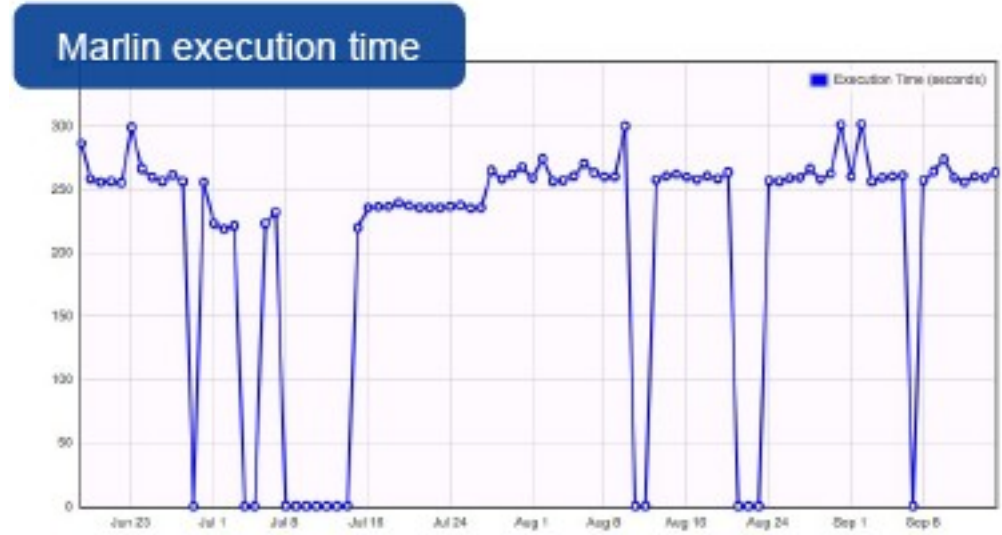
Site	Build Name	Update		Configure			Build			Test				Build Time
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min	
grid-llc-pa0	linux-gcc-debug			0	0	0	0	12	0.1					2010-09-15T02:01:57 CEST
grid-llc-pa0	linux-gcc-debug-x64			0	0	0	0	12	0.1					2010-09-15T04:01:42 CEST
grid-llc-pa0	linux-gcc-default			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

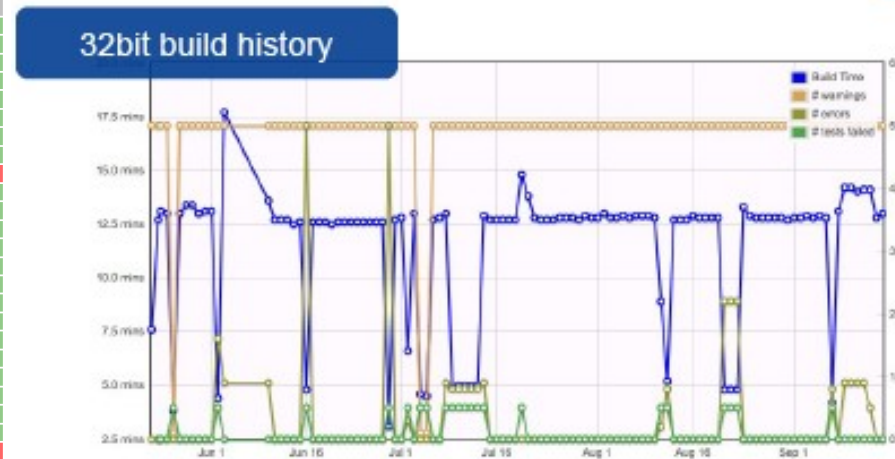
new development: ILCTest

S.Aplin, J.Engels

- dashboard provides:
 - detailed statistics on test performance:
 - Errors, Warnings,...
 - build times
 - history plots
 - access to test output (per package and per test)



History								
Date	Update Files	Update Errors	Update Warnings	Configure Errors	Configure Warnings	Build Errors	Build Warnings	Tests Failed
2010-09-15 03:01:15	0	0	0	0	0	0	50	0
2010-09-14 03:01:14	0	0	0	0	0	0	50	0
2010-09-13 03:01:13	0	0	0	0	0	9	50	0
2010-09-12 03:01:12	0	0	0	0	0	9	50	0
2010-09-11 03:01:11	0	0	0	0	0	5	50	0
2010-09-10 03:01:10	0	0	0	0	0	9	50	0
2010-09-09 03:01:09	0	0	0	0	0	9	50	0
2010-09-08 03:01:08	0	0	0	0	0	0	50	0
2010-09-07 03:01:07	0	0	0	0	0	8	50	5
2010-09-06 03:01:06	0	0	0	0	0	0	50	0
2010-09-05 03:01:05	0	0	0	0	0	0	50	0
2010-09-04 03:01:04	0	0	0	0	0	0	50	0
2010-09-03 03:01:03	0	0	0	0	0	0	50	0
2010-09-02 03:01:02	0	0	0	0	0	0	50	0
2010-09-01 03:01:01	0	0	0	0	0	0	50	0
2010-08-31 03:01:31	0	0	0	0	0	0	50	0
2010-08-30 03:01:30	0	0	0	0	0	0	50	0
2010-08-29 03:01:29	0	0	0	0	0	0	50	0
2010-08-28 03:01:28	0	0	0	0	0	0	50	0
2010-08-27 03:01:27	0	0	0	0	0	0	50	0
2010-08-26 03:01:26	0	0	0	0	0	0	50	0
2010-08-25 03:01:25	0	0	0	0	0	0	50	0
2010-08-24 03:01:24	0	0	0	0	0	0	50	0
2010-08-23 03:01:23	0	0	0	0	0	22	50	5
2010-08-22 03:01:22	0	0	0	0	0	22	50	5
2010-08-21 03:01:21	0	0	0	0	0	22	50	5
2010-08-20 03:01:20	0	0	0	0	0	0	50	0
2010-08-19 03:01:19	0	0	0	0	0	0	50	0



new development: ILCTest

S.Aplin, J.Engels

- for unit tests:
 - simple test class + CMake macros
 - easy to write small test programs and automatically run them with cmake/ctest
- for physics tests:
 - started work with summer student
 - program for checking histograms and comparing them to reference histograms
 - Kolmogorov, Chi2,....
 - should make it easy for every group to add their favorite physics tests

```
// 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!") ; }
```

Calice has started to use ILCTest

- run test in Nightly builds
- monitor software quality

→ encourage other groups to join

EUTelescope

I.Rubinsky

- framework 'declared final' – recent activities:
- improved user convenience:
 - Python scripts for configuration and submission
 - optional transparent submission to the Grid
- greatly improved CPU performance (O(10-100))
- wish list for core software
 - need tilted sensors in GEAR – misalignment ! ← to be addressed soon
 - would like to have LCIO file browser → use new LCIO ROOT dictionary

quote: I. Gregor JRA1 summary:

• What would we do differently?

• Use stand-alone analysis software (Marlin is a difficult for non ILC users).

→ not so surprising :-)

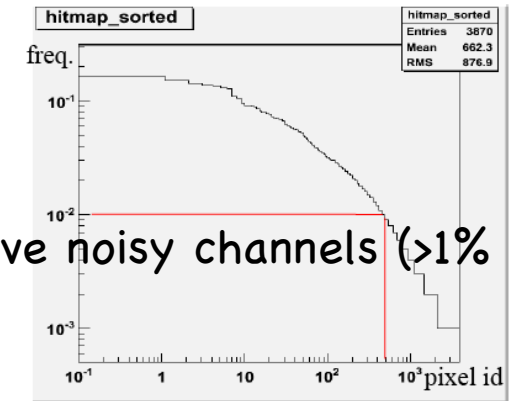
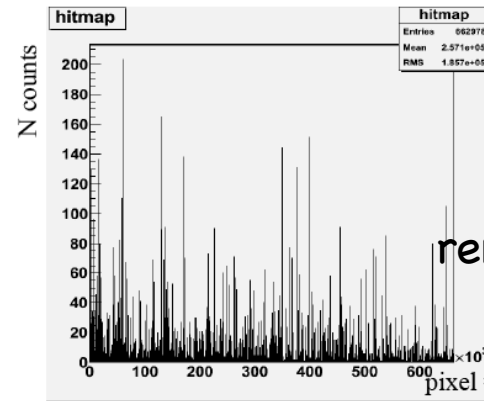
Modular **A**nalysis & **R**econstruction for the **L I N**ear Collider

but seriously: not convinced that writing a standalone analysis framework would have been more efficient or convenient for the users and developers !

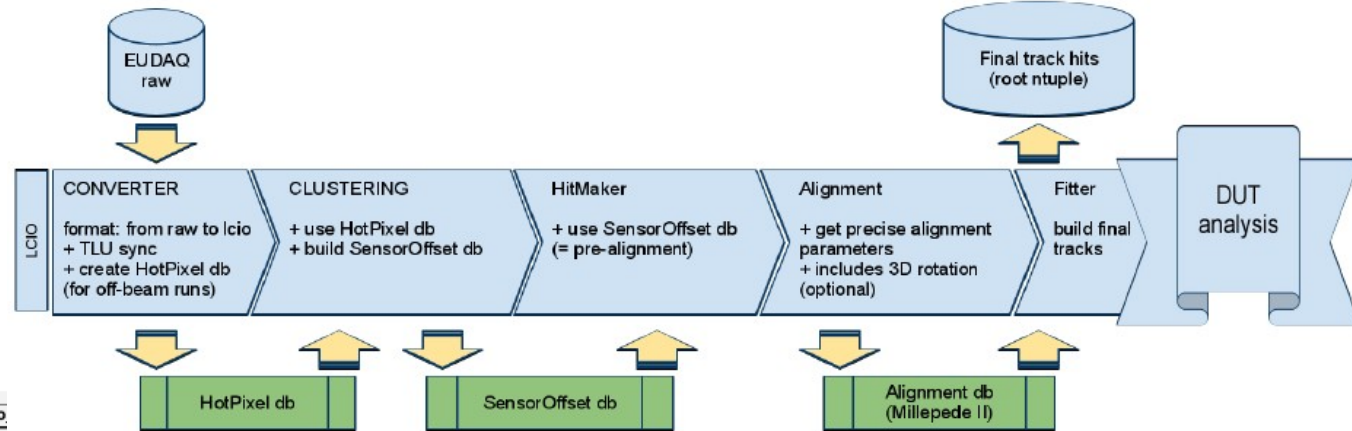
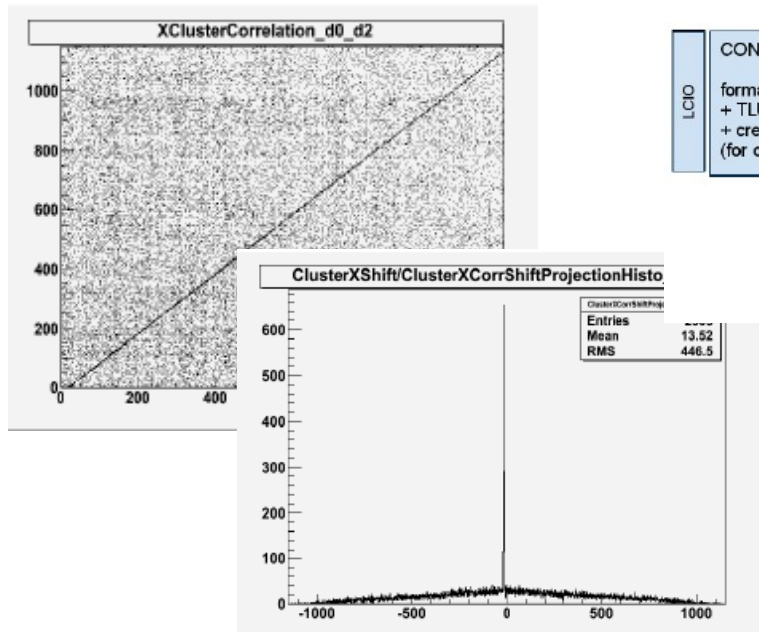
EUTelescope

I. Rubinsky

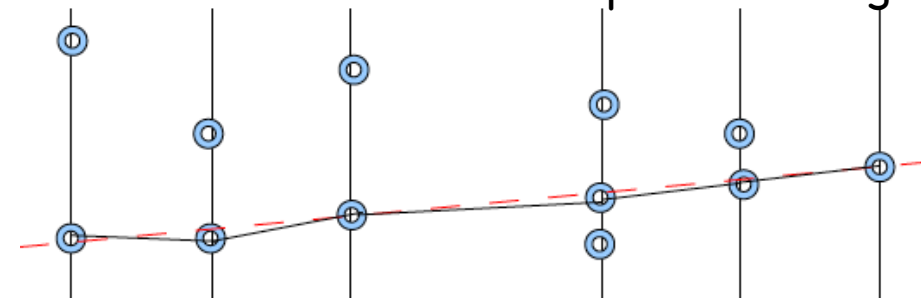
- improved CPU performance:
 - clustering
 - pre-alignment
 - fitter



remove noisy channels (>1% occup.)



use relative beam slope for fitting



pre-align on pixel position (~1/2 pixel pitch)

MarlinTPC Project Overview

International effort with participation from all regions (Americas, Asia, Europe)

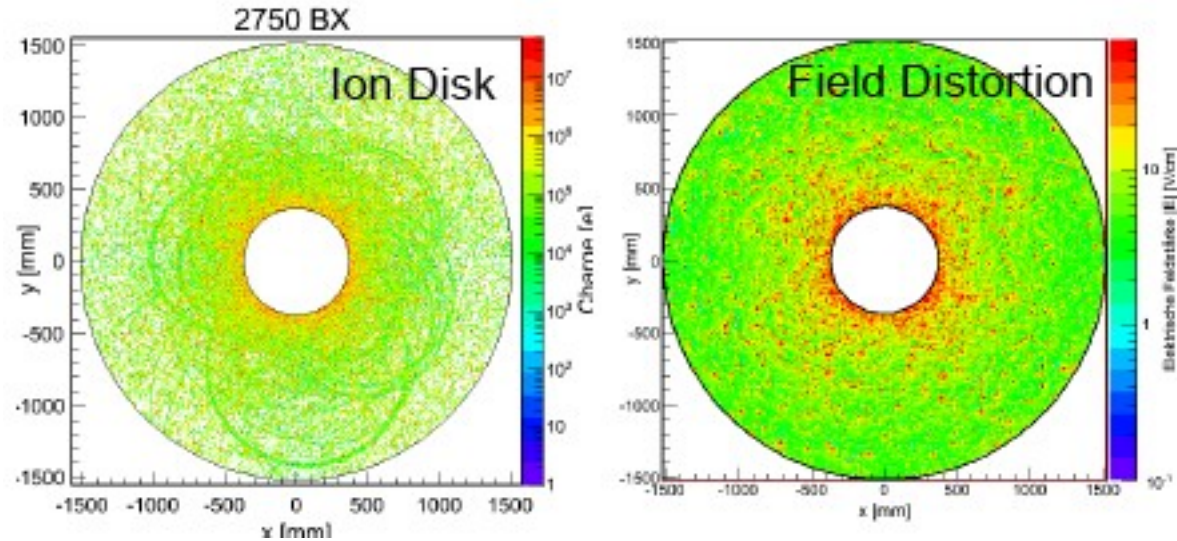
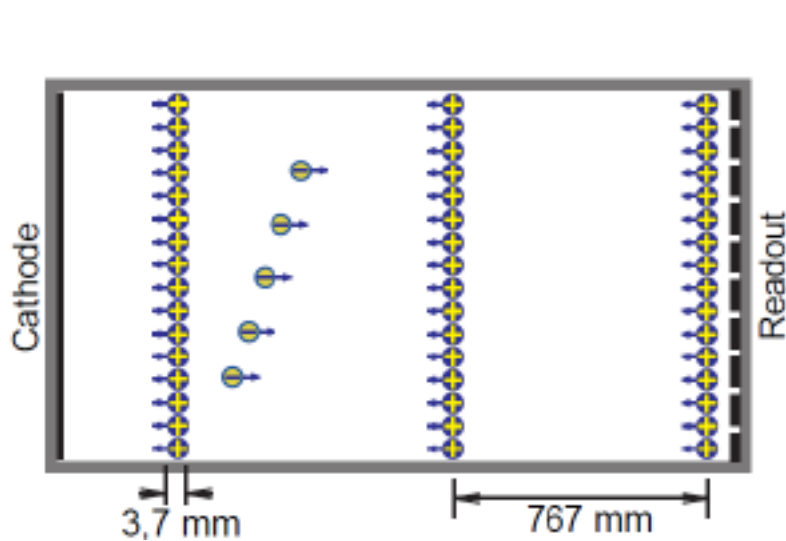
Software uses the ILC software framework (MARLIN, GEAR, LCCD)

Modular software design allows to compare algorithms and reuse existing solutions

- recent activities :
- restructured the code
 - loose binding to Marlin processors
 - → create multi use libraries
- much of functionality just became available during last year
 - Kalman Fitter implemented
 - Ion back drift simulation
 - first LCTPC test beam data processed in MarlinTPC

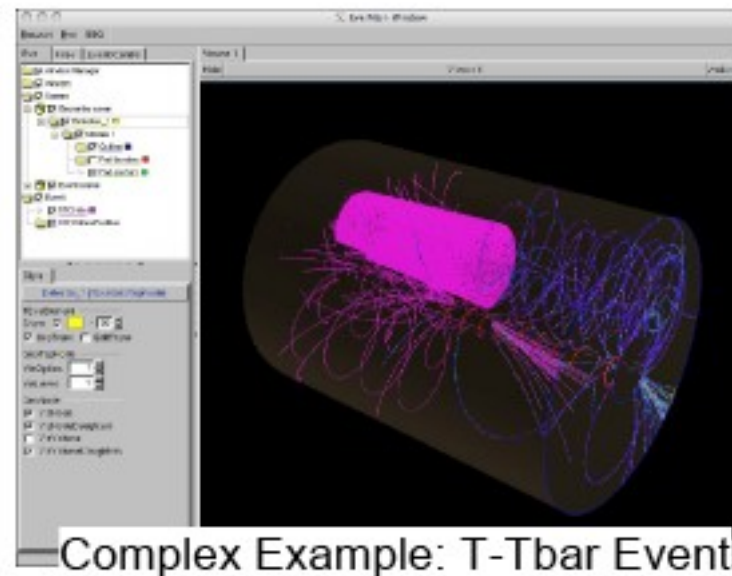
LCIO Type	Default Collection Name
TrackerRawData	TPCRawData
TrackerData	TPCConvertedRawData
TrackerPulse	TPCPulses
TrackerHit	TPCHits
Track	TPCTracks

Ion back drift simulation in MarlinTPC: study the field distortions



- Summer student project
- TEve based:
 - C++, ROOT
 - OpenGL: 3D hardware acceleration
 - Object browser
 - Picking and highlighting
- Full GEAR support
 - All pad layouts
 - Multiple modules
- LCIO support

new event display for MarlinTPC/LCTPC



Complex Example: T-Tbar Event

implemented Kalman Filter based track fitting (and finding) for LCTPC

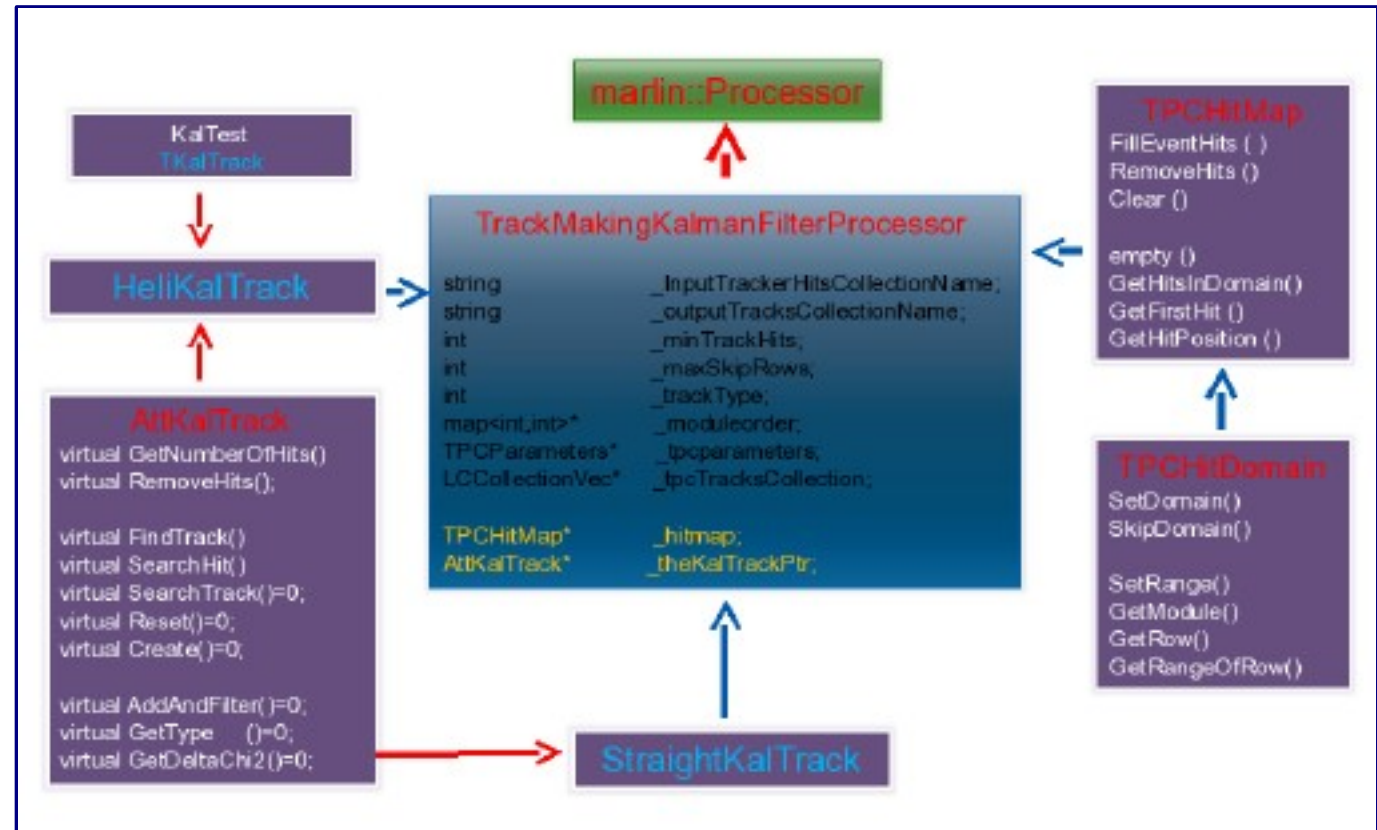
Kalman Fitter library ready

Source committed to MarlinTPC repository

Can fit tracks over several modules

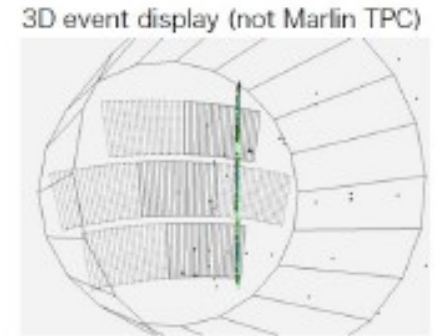
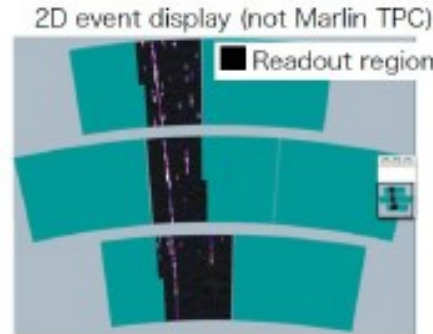
Track finding possible but currently still limited

Planned: implement solution to fit tracks in non-uniform magnetic fields



- code uses KalTest package (K.Fuji et al)
- to be included in next iLCSoft release
- also to be used by new tracking code for ILD !

- (Very) Preliminary test beam results from Asian colleagues:

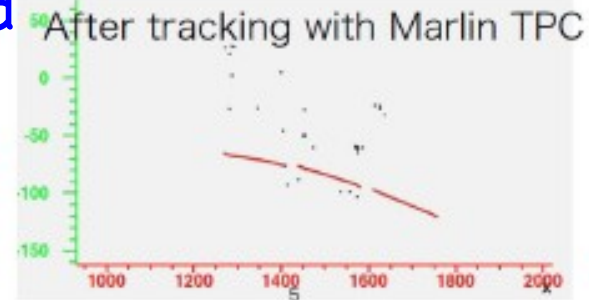


- Use of a complete MarlinTPC chain in reconstruction

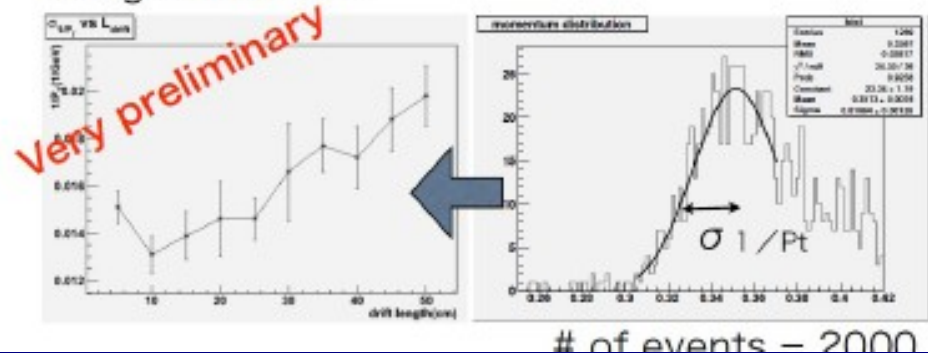
- Planned:

first data reconstructed with MarlinTPC !

- Correction of module alignment
- Correction of field distortion (E-field near GEM)
- Tracking in non-uniform B-field
- Shift to MarlinTPC for analysis



Momentum resolution ($1/Pt$) without any alignment correction Using Marlin TPC



- first community to adapt the iLCSoft framework
 - LCIO, Marlin, LCCD – even before start of EUDET
 - conversion of raw data to LCIO
- wish list for core software:
 - user defined data classes in LCIO ← needs discussion and thought

Test beam campaigns

- DESY 2006: AHCAL
- CERN 2007: Si-W ECAL + AHCAL + TCMT
- FNAL 2008: Si-W ECAL + AHCAL + TCMT
- FNAL 2009: Si-W ECAL + AHCAL + TCMT,
scintillator ECAL + AHCAL + TCMT
- CERN 2010: W-HCAL (ongoing)
- FNAL 2010: DHCAL (in preparation)

- all central data processing on the Grid
- 30 TB data stored at DESY SE (replicated to Lyon Grid)
- adopted new ILCTest system
- main focus on data analysis → previous talk by E.Garutti

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

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

LC related deliverables and milestones

D2.2.1	Software design for geometry toolkit including the interfaces for the reconstruction toolkits	CERN, DESY, LLR, UniGla, STFC	M12
D2.2.2	Software toolkit for detector geometry, materials and detection technologies		M38
D2.3.1	Software design for tracking toolkit	DESY, OeAW, KFKI	
D2.3.2	Software design for PFA tools	Ucam, LLR, CERN,	M12
D2.3.3	Design for handling the pile-up in sLHC	INFN, NTU, KFKI	M18
D2.3.4	Software toolkit with tracking algorithms	DESY, OeAW, KFKI	
D2.3.5	Particle Flow software tools	Ucam, LLR, CERN,	M38

deliverables and milestones with contributing partners

MS10	Running first prototype of the particle flow algorithm.	UCAM, LLR	Application to LC detector (Task 2.3)
MS11	Running prototype of tracking toolkit including some algorithms	DESY, OeAW	Application to ILD-TPC simulation (Task 2.2)
MS12	Running prototype of the geometry toolkit	CERN, DESY, LLR	Application to ILD detector simulation (Task 2.2)
MS13	Running prototype of the tracking code for the pile-up	INFN, NTU, KFKI	Application to sLHC simulation (Task 2.3)
MS14	Integration of tracking toolkit into LC software framework	DESY	Validation of physics performance (Task 2.3)

first deliverables focus on **software design**:

need to properly define the (abstract) interfaces

milestones involve integration in existing LC framework

next steps: need to find best way to organize the collaboration among partners, e.g. regular phone meetings, occasional face to face meetings...

AIDA WP2 and LC test beam

- in EUDET we have successfully integrated the software requirements for the test beam communities with the activities for the full detector simulation and optimization
- also the LC activities in AIDA WP2 are mainly targeted at the detector concepts we will continue to tightly collaborate with the test beam groups
- e.g.:
- the new geometry system needs to address the requirements for test beam setups (as in GEAR)
- ideally integrate geometry and conditions data (validity time intervals)
- also alignment task (targeted at sLHC) should be of interest for the test beams

Summary & Outlook

- task ANALYS in EUDET targeted at core software for data processing of test beam experiments for the LC
- integrated with common iLCSoft framework also used by ILC and CLIC
- all three JRAs have fully adopted the common software and produced very nice (first) results
- the Grid is used for most of the centralized data processing
- in AIDA WP2 we will develop general purpose tools for future HEP experiments with a strong focus on Linear Collider
- main target is geometry and reconstruction (tracking & PFA)
- will continue to fully support test beam experiments throughout AIDA
– and beyond the scope of AIDA