Preparation of ilcsoft v01-17-07

F. Gaede

DESY/CERN

ILD Software and Analysis Meeting
May 27, 2015

Outline

1. Introduction
2. Core Tools
3. Tracking Tools
4. Calorimeter Reconstruction (PFA)
5. High Level Reconstruction Tools
6. Other Developments
7. Timeline and Discussion

Introduction

- next in series of developers releases, started with v01-17-01 after DBD
- last version v01-17-06 released in August 2015
- since then many new developments:
  - new DD4hep based simulation
  - Tracking
  - PandoraPFA
  - High Level Reconstruction and Analysis Tools
  - Digitizers
  - ...
- this talk:
- focus on major developments and implications
- start discussion about timeline

Preparing for using C++11

- start to allow usage of C++11 with this release
- allows developers to benefit from many new features in C++
- required to benefit from major improvements in PandoraPFA
  - (see talk by J.Marshall)
- required by ROOT 6 (not yet in this release)
- for now we simply add `-std=c++11` to all packages

Implication

- need to use non-system compiler on SL6 (SL5) for all packages
- need sufficently recent compiler ($>$ gcc4.6, clang5) on other platforms

gcc 4.8 on SL6 (SL5)

- using non-system compiler on SL6 (ships with gcc 4.4) requires:
- that all packages are built with this compiler
- shipping of this compiler with ilcsoft and/or ilcinstall
- to save duplication of effort use the gcc 4.8 provided by CERN SFT group in afs and on cvmfs
- before installation/compilation, one needs to run (SL6):

```
source /afs/cern.ch/sw/lcg/external/gcc/4.8.1/x86_64-slc6-gcc48-opt/setup.sh
export PATH=/afs/cern.ch/sw/lcg/external/Python/2.7.4/x86_64-slc6-gcc48-opt/bin/:$PATH
export LD_LIBRARY_PATH=

/afs/cern.ch/sw/lcg/external/Python/2.7.4/x86_64-slc6-gcc48-opt/lib/:$LD_LIBRARY_PATH
```

- SL5:

```
source /afs/cern.ch/sw/lcg/contrib/gcc/4.8.0/x86_64-slc5-gcc48-opt/setup.sh
export PATH=/afs/cern.ch/sw/lcg/external/Python/2.7.2/x86_64-slc5-gcc46-opt/bin:$PATH
export LD_LIBRARY_PATH=
/afs/cern.ch/sw/lcg/external/Python/2.7.2/x86_64-slc5-gcc46-opt/lib:$LD_LIBRARY_PATH
```

- currently in test phase w/ nightly builds

boost

- DD4hep (DDG4) needs boost header files
- so far relied on pre-installed boost
- now also provide boost header files (independent of OS) at `/afs/desy.de/project/ilcsoft/sw/boost/1.58.0`
- also to be used in
  - LCFIVertex (optionally replaces the old internal boost version)
  - LCFIPlus will automaticall pick up this external boost

Implication

- all other ilcsoft packages are free to use boost as well
- provided they use header-only packages !

DD4hep

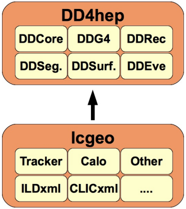
- about to prepare new DD4hep release (v00-12) which should provide
- complete functionality for running simulation and reconstruction
- many changes since las release v00-11, e.g.
  - DDRec interface to eventually replace GEAR
  - SurfaceManager to access surfaces as needed for tracking
  - updated to Geant4 10.x series (requires at least Geant4 9.6)
  - started preparation for ROOT 6 (next release !?)
  - updated to optionally use C++11
  - introduced component structure: only link against what you use
  - lots of fixes and improvements

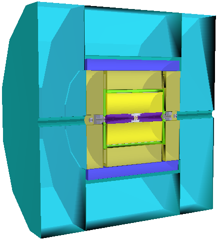
Implication

will not be able to provide iLCSoft release with DD4hep and Geant4 9.5

lcgeo

- lcgeo (f.k.a. DDSim): common LC detector description package for ILD and CLIC
- use simply python script for configuring and running the simulation: ddsim.py
- ported current Mokka model ILD_o1_v05 to DD4hep
  - can start validation (soon)
- introduced 'mandatory' envelopes
- detector geometry is loaded as a plugin at runtime





Implication

can update lcgeo (geometry) without having to recompile any other package

Marlin and DD4hep

- currently Marlin depends on EDM and geometry, i.e. LCIO and GEAR
- try to avoid an additional dependency on DD4hep:
- created small standalone Marlin package: MarlinDD4hep
- this allows to have Marlin packages that do not need the geometry to not have to link against DD4hep

Implication

every Marlin application that uses DD4hep, needs to run the InitializeDD4hep processor as first processor in the steering file

KalTest

- new version of KalTest that can fit in inhomogeneous B-fields exist (Li Bo)
- can be used transparently also for homogeneous B-field w/o field map
- moved to trunk for testing
- observe issues with quality of track fit (pull distributions)
  - under investigation

Implication

- issues need to be understood and resolved
- can this be done for this release ?
- otherwise we will have to revert to old and tested version

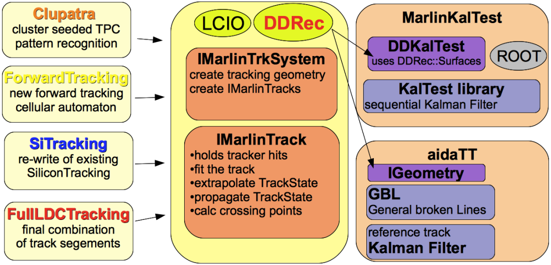
DDKalTest

- implementation of measurement surface and hit classes needed for KalTest Kalman filter used in MarlinTrk
- re-implement some classes from KalDet that used GEAR to instantiate the surfaces - now using the DDRec::Surfaces from the DD4hep model
- no GEAR file is needed
- DD(Parallel)PlanarMeasLayer
  - planar measurement layers (parallel and orthogonal to z)
  - works for 1D and 2D hits: VXD, SIT, SET, FTD, (all-silicon-tracker)
- DDCylinderMeasLayer
  - cylindrical measurement layers parallel to z: TPC

Implication

with DDKalTest we can run the KalTest Kalman filter on any tracking detector that has the `DDRec::Surfaces` implemented
no additional glue code needed!

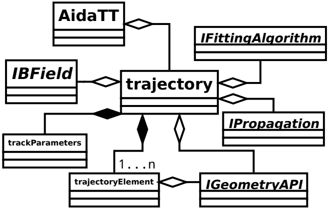
MarlinTrk



- added MarlinDDKalTest and MarlinDDKalTestTrack implementing the IMarlinTrk interface
- changed factory to provide one instance of the tracking system per type
- updated all tracking processor to allow for chooosing the fitting type:
  - SiliconTracking, ForwardTracking, Clupatra, FullLDCTracking, Refitting, ...

aidaTT

- generic tracking tool developed in AIDA-WP2
- implementation of GBL exists
- testing and inclusion of material effects ongoing
- next step: implementation of MarlinTrk interface
  → ongoing



Implication

aidaTT provides an alternative fitter - not strictly needed for next round of ILD optimization

Calorimeter Reconstruction (PFA)

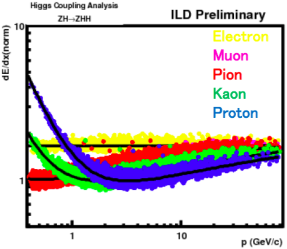
- PandoraPFA: many improvements w.r.t physics and computing performance (see talk J.Marshall)
  - will be included in v01-17-07
- Garlic: higher γ efficiency and 2-γ separation
  - to be included in v01-17-07 !?
- Arbor: independent PFA implementation
  - is this included in PandoraPFA framework
  - can it be run parallel to PandoraPFA ?
- π^0 Finder: using MVA
- BeamCalReconstruction: improved efficiencies

Implication

- need to understand status and timeline of tools for this release
- also need to understand interplay with PandoraPFA
  - ideally want a unique list of PFOs

High Level Reconstruction Tools

- lots of progress in high level tools
- LCFIPlus - flavor tagging
- Improvement of PID
  - e.g. using dE/dx in the TPC
- Vertex Charge
- many more (see talk J.List/J.Tian)
- tools still under active development



M.Kurata et all

Implication

- tools should be used for ILD detector optimization
  - possibly also for a re-DSTing of existing REC files
- need to make an effort to finalize tools to include in v01-17-07
- have mini-workshop at DESY for this ? (see talk J.List/J.Tian)
- → timeline ?

Other Developments

- many additional new developments not covered in this talk , e.g: improved calorimeter digitizers and kinematic fitting, FastJetContrib package,...

Note to Authors

- if your latest developments are in one of the existing centrally maintained iLCSoft packages:
  - please make sure to add a few lines to the release notes of the corresponding package such that your work gets acknowledged
- if you plan to have a new (version of your) package included in iLCSoft v01-17-07
  - let us know asap about your package and timeline
  - prepare a tagged version of your package
  - build and test against a current pre-release

Pre-releases at

```
/afs/desy.de/project/ilcsoft/sw/x86_64_gcc48_sl6/v01-17-07-preXY
```

Timeline and Discussion

- the v01-17-07 iLCsoft release will bring a very large number of changes and new features and addresses two main purposes:
- serve as start release for the validation of the new DD4hep based software tools (for ILD and CLIC)
- provide released versions of many new (high level) reconstruction and analysis tools for physics analysis and detector optimization

Timeline

- need to understand the timelines for these two aspects
- for the new software tools we would like to have the release asap (middle/end of June ?)
- for the reconstruction tools it might take a bit longer ...
- should in any case aim for a release before summer (end July)
- possibly have one earlier release followed by a smaller patch release
- open for discussion ...