DD4hep Based Event Reconstruction

Markus Frank*, Frank Gaede†, Daniel Hynds*, Shaojun Lu†, Nikiforos Nikiforou*, Marko Petric*, André Sailer*, Rosa Simoniello*, Georgios Voutsinas†

* CERN, † DESY, Hamburg

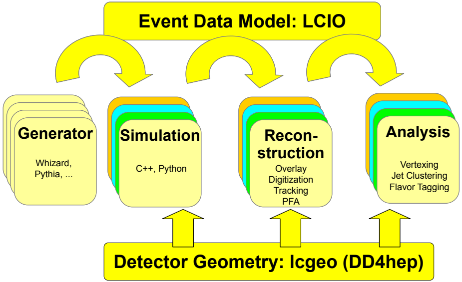
On behalf of the CLICdp and ILD collaborations

Computing in High Energy Physics
San Francisco, California, USA
October 10–14, 2016

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.

Introduction

- Linear collider community has used and developed **common software** for many years
  - Event data model: LCIO [1, 2]
  - Particle flow reconstruction: PandoraPFA [3]
- Adopted DD4Hep geometry description to develop more common software based on the DD4Hep geometry information
- Needs to be usable ***at least*** by all linear collider detector collaborations
  - Interface generic reconstruction packages via thin wrappers to linear collider framework



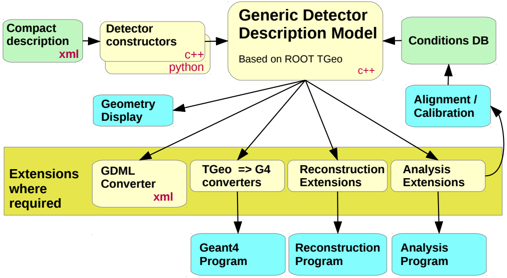
DD4HEP Geometry Framework

DD4HEP is the **single source** of geometry information for simulation, reconstruction, analysis

- Avoid duplicate implementation of geometry for different purpose or inconsistencies are almost certain

Further information:

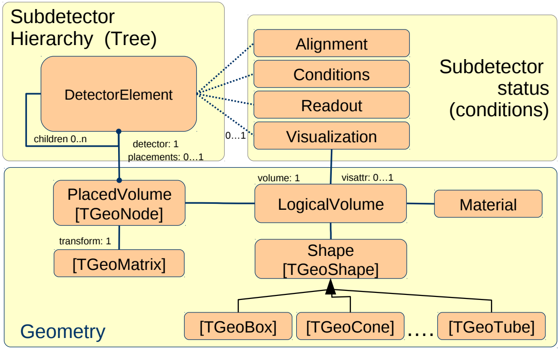
- Presentation on Detector Simulation with DD4hep
- DD4hep framework [4]
- DDG4: The DD4HEP gateway to Geant4 [5]



Geometry Tree

In DD4HEP detectors are a tree of `DetElements`

- A `DetElement` is a sub-detector or a significant part of a sub-detector
- `DetElement` points to placed logical volumes
- Extensions can be attached to `DetElements` to **provide additional views of the geometry information**



Detector XML Example

- XML structure to set parameters for detectors
- `C++` driver to interpret XML parameters and create `DetElements` and `Volumes`
  - Define sensitive parts (attached with `SensitiveDetector`) and radiator, which has to be known for reconstruction purposes
- Attach sensitive detector readout, defined elsewhere in XML

```
<readout name="ECB">
  <segmentation type="CartesianGridXY"
                grid_size_x="ECal_cell_size"
                grid_size_y="ECal_cell_size" />
  <id>...x:32:-16,y:-16 </id>
</readout>
```

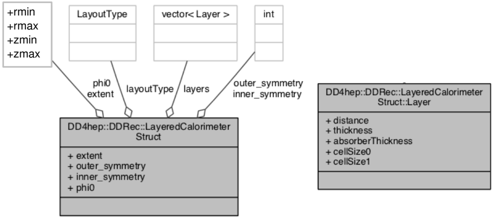
```
<detector
    name="ECalBarrel"
    type="GenericCalBarrel_o1_v01"
    id="42" readout="ECB">
  <dimensions
      numsides="ECalBarrel_symmetry"
      rmin="ECalBarrel_inner_radius"
      z="ECalBarrel_half_length*2" />
  <layer repeat="25" >
    <slice material="Tungsten"
           thickness="2.40*mm"
           radiator="yes"/>
    <slice material="Air"
           thickness="0.25*mm" />
    <slice material="Silicon"
           thickness="0.50*mm"
           sensitive="yes"/>
  </layer>
</detector>
```

DDREC: High Level Information

High level view onto the detectors through DDREC `DataStructures` extensions for `DetElements`

- C++ driver fills DDREC `DataStructures`
- Dimensions, positions, parameters for layers,. . .
- Reconstruction accesses information from `DataStructures`. **Decouples detector implementation from reconstruction algorithms**
- Detector drivers responsible to fill and attach appropriate `DataStructures` to sub-detectors

`DataStructures` contain sufficient information to provide geometry information to particle flow clustering via PANDORAPFA



Data Structure	Detector Type
`ConicalSupportData`	Cones and Tubes
`FixedPadSizeTPCData`	Cylindrical TPC
`LayeredCalorimeterData`	Sandwich Calorimeters
`ZPlanarData`	Planar Silicon Trackers
`ZDiskPetalsData`	Forward Silicon Trackers

Geometry for Track Reconstruction

Information needed for track reconstruction

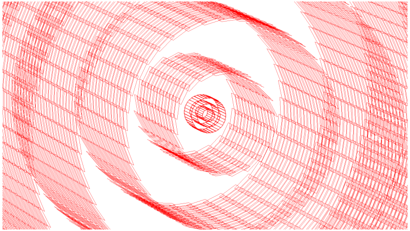
- measurement directions of hits
- local-to-global coordinate transforms
- material properties

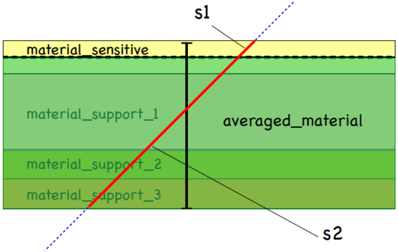
DD4HEP surfaces provide this information

- Surfaces can **automatically** added (plugin) or explicitly in the driver

```
<plugin name="DD4hep_GenericSurfaceInstallerPlugin">
  <argument value="TrackDet"/>
  <argument value="dimension=2"/>
  <argument value="u_x=-1."/>
  <argument value="v_y=-1."/>
  <argument value="n_z=1."/>
</plugin>
```

  - Plugin loops over all `DetElements` of given sub-detector and adds surfaces for sensitive elements
  - Configure surface type and direction in volume
- Surfaces automatically average materials





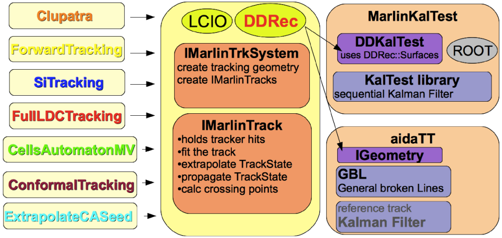
Track Reconstruction

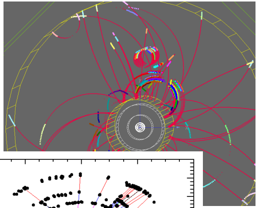
Track reconstruction for linear collider software becoming more **framework independent** by using DD4HEP surfaces

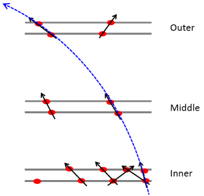
- Pattern recognition/track finding algorithms
  - From **detector specific**: *Clupatra* for TPC; mini-vector for vertex detector double layers
  - to **geometry agnostic**: pattern recognition in conformal space
- Track fitting, fairly generic: DDKALTEST, AIDATT
- Geometry: Interfaced via DDREC and Surfaces

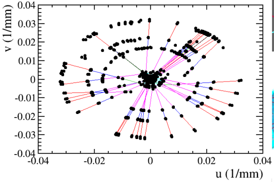
DDKALTEST using DD4HEP surfaces for track fitting

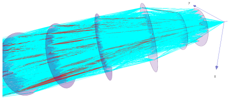
- **If detector is described in DD4HEP track reconstruction will run on it**











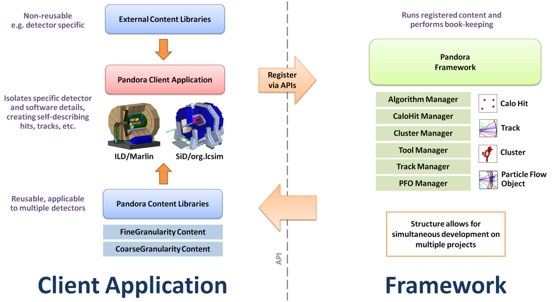
Particle Flow Reconstruction

PANDORAPFA: generic toolkit for pattern recognition algorithms in highly granular calorimeters

- Originally developed for ILC/CLIC detectors
- Extended to work in LAr-TPC reconstruction for the DUNE neutrino experiment
- Some functionality for *coarse* calorimeters exists as well

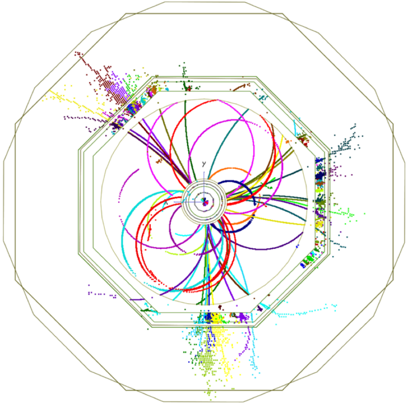
ClientApplication: DDMARLINPANDORA glues linear collider framework (Marlin), DD4HEP, and PANDORAPFA

- Passes DDREC `DataStructures` information, tracks, and calorimeter hits to PANDORAPFA
- Converts PANDORAPFA objects into LC EDM objects

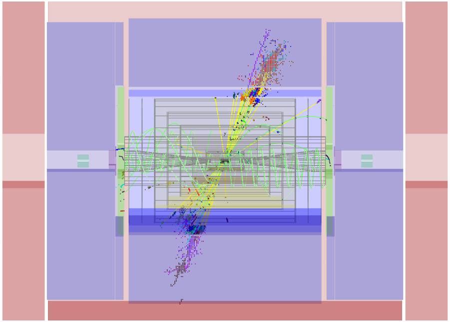


Reconstructed Events

Two events in **different** detectors **simulated and reconstructed** with previously described tracking and particle flow clustering



ILD detector, TPC and silicon tracking: $t\,\bar{t}$



CLIC detector, silicon tracking: Z to two jets

Summary

The DD4HEP detector description tool-kit offers a flexible and easy to use solution for the consistent and complete description of particle physics detectors in one single system.

- No need to define separate geometry for reconstruction, no duplication of effort, fewer bugs
- Generic reconstruction packages with *no* framework dependency: tracking toolkit (aidaTT), particle flow reconstruction (PANDORAPFA)

If the detector is described via DD4HEP, reconstruction comes almost for free

- Used by CLIC, ILD, SiD for simulation and **reconstruction**
- Only need a thin wrapper between experiment framework and geometry on one side and reconstruction package on the other
- Well suited for detector optimisation studies at other future accelerators

References

[1] F. Gaede, T. Behnke, N. Graf, and T. Johnson. LCIO - A persistency framework for linear collider simulation studies. In *Proceedings of CHEP 2003*. 2003.

[2] F. Gaede, et al. LCIO persistency and data model for LC simulation and reconstruction. In *CHEP 2004*. 2005.

[3] J. S. Marshall and M. A. Thomson. The Pandora software development kit, CHEP 2012. In *CHEP 2012*, p. 022034. 2012. AIDA-CONF-2015-003.

[4] M. Frank, F. Gaede, and P. Mato. DD4hep: A Detector Description Toolkit for High Energy Physics Experiments. In *CHEP 2013*. 2013. AIDA-CONF-2014-004.

[5] M. Frank, F. Gaede, N. Nikiforou, M. Petric, and A. Sailer. DDG4 A Simulation Framework based on the DD4hep Detector Description Toolkit. In *CHEP 2015*. 2015.