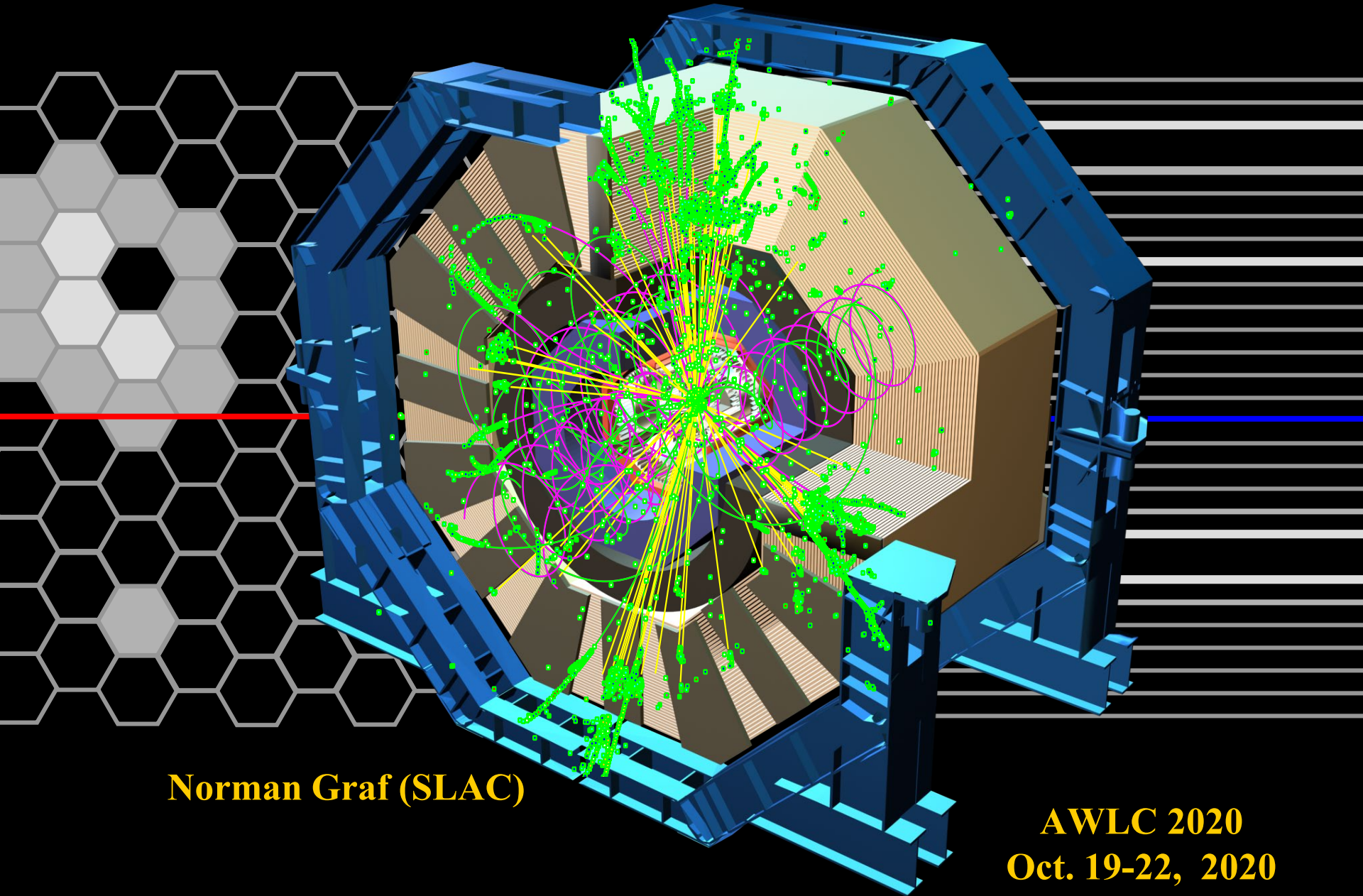


# *Full Simulation & Reconstruction*



**Norman Graf (SLAC)**

**AWLC 2020  
Oct. 19-22, 2020**

# Full Detector Simulation Goals

- Fast simulations are useful tools to quickly investigate new physics processes or analyses, but full simulation of the detector response is crucial to the design of real detectors.
- Enables the simulation of realistic effects which are not usually addressed using fast simulation tools.
  - Interactions in tracking detectors (conversion, showering)
  - Full showering in calorimeters, including albedo in trackers
  - Electronic response simulations, including noise & pileup
  - Digitization and “ghost hit” effects in discrete detectors
  - Pattern recognition (track finding & calorimeter clustering)
  - ...

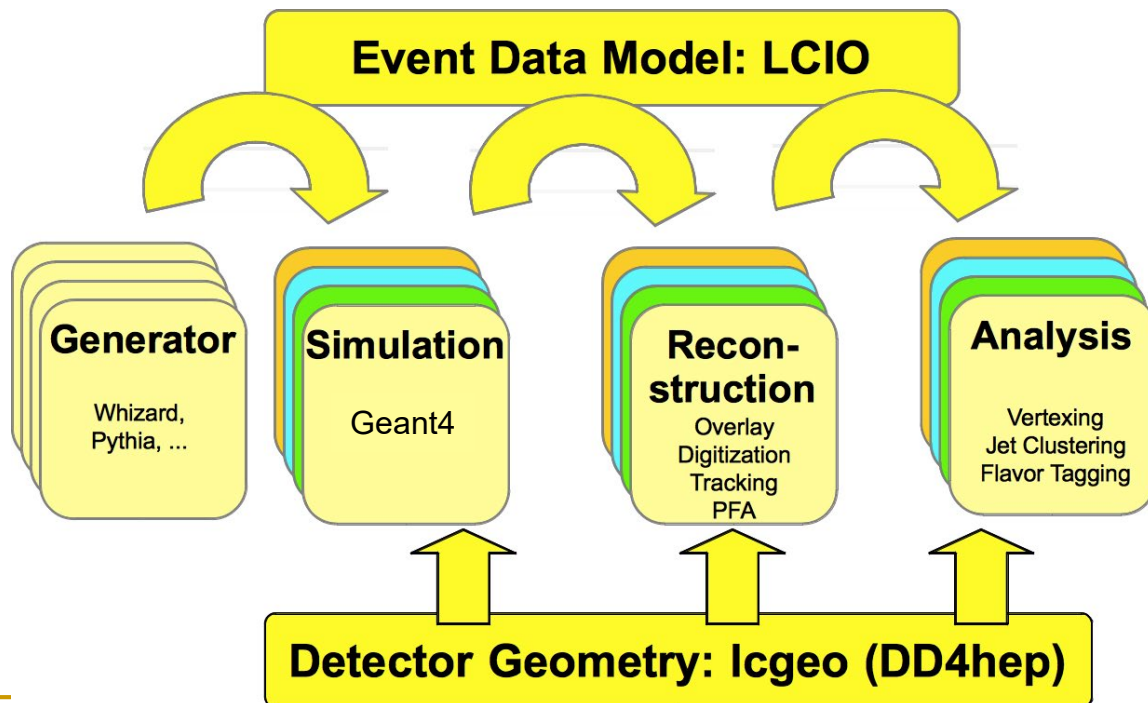
---

# iLCSoft Goals

- Facilitate contribution from physicists in different locations with various amounts of available time.
- Use standard data formats, collaborate & cooperate.
- Provide a general-purpose framework for physics simulation, reconstruction and analysis software development.
- Develop a suite of reconstruction and analysis algorithms and sample codes.
- Simulate benchmark physics processes on different full detector designs.
- Deliver software which is easy to install, learn, use.
- Support via web based forums, tutorials, meetings.

# iLCSoft Overview

- Developed as a common software framework for ILC detector studies (ILD & SiD)
  - also used by CLIC, Calice, LCTPC (FCC, CEPC, MuC, EIC, . . .)
- Key components
  - LCIO: common event data model (EDM) and persistency
  - DD4hep: common detector geometry description



# Full Simulation Data Flow

## Geant4 outputs

MCParticles

Generator Input + Secondaries from interactions

SimTrackerHits

x, t, edep, mcp from G4step in sensitive tracker volume

SimCalorimeterHits

Aggregated edep, t, mcp in cell volume

## Electronic response

Channel ID, FADC pulse, or ADC counts

Includes readout noise / pileup

## Digitization

Conversion of electronic signals to position and energy, time, etc.

## Particle Reconstruction

Pattern recognition and fitting of tracks,

Clustering and identification of calorimeter hits

Association of tracks and clusters to particles

## Higher-level Reconstruction

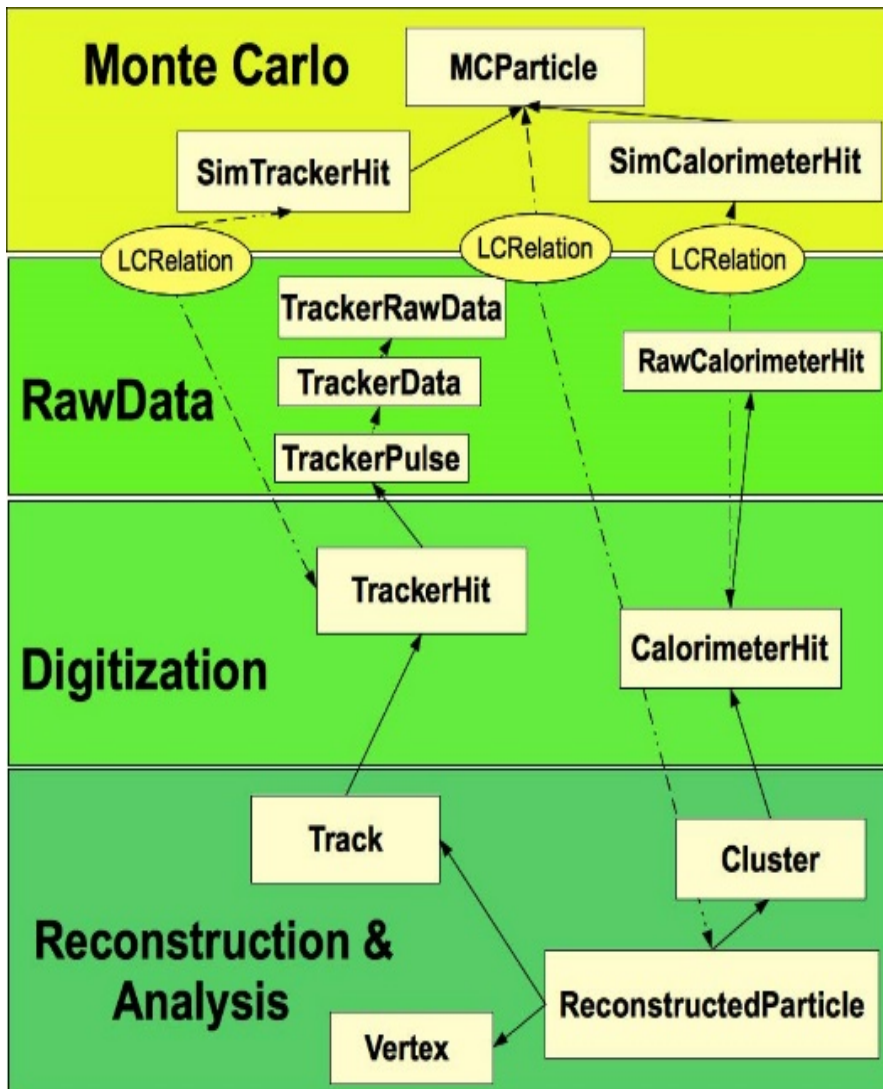
Jet Finding

Vertexing

Flavor tagging

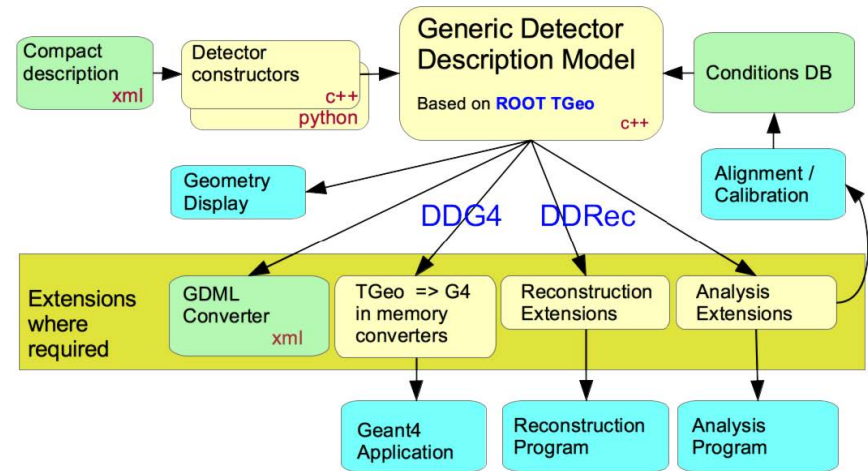
## Monte Carlo Truth

LCRelation collections provide the links between reconstructed objects and the MCParticles which produced their signals.



# DD4hep

- *Detector Description for HEP* is the common detector geometry description for iLCSoft
- Same detector model is used for:
  - simulation,
  - reconstruction
  - visualization
  - analysis
- Detector fully described via:
  - C++ detector constructors
  - XML files (*compact files*)



**DDG4** full simulation  
with Geant4

**DDRec** interface for  
reconstruction

---

# Marlin Event Reconstruction

- C++ application framework used throughout iLCSoft
- Tasks are implemented as Processors
  - Full Event Reconstruction
    - Beam background overlays at detector hit level, including time offsets.
    - Detector readout digitization (1D Si  $\mu$ -strips, 2D Si pixels, 3D TPC pad hits)
    - *ab initio* track finding and fitting for  $\sim$ arbitrary geometries
    - Multiple calorimeter clustering algorithms
    - Individual Particle reconstruction (cluster-track association)
    - Jet Finding, Vertex Finding, Flavor Tagging
- Applications fully configured via XML files
  - global parameters
  - the chain of processors to run
  - per processor parameters
- <http://ilcsoft.desy.de/Marlin/current/doc/html/index.html>

# Marlin Steering Files

## Define processors to be run

```
<execute>
<processor name="MyAIDAProcessor"/>
<processor name="InitDD4hep"/>
<processor name="VXDPlanarDigiProcessor"/>
<processor name="SITPlanarDigiProcessor"/>
<processor name="SITDDSpacePointBuilder" />
<!-- ... -->
<processor name="DSTOutput"/>
</execute>
```

## Define global parameters

```
<global>
<parameter name="LCIOInputFiles"> bbudsc_3evt_SIM.slcio </parameter>
<parameter name="MaxRecordNumber" value="0"/>
<parameter name="SkipNEvents" value="0"/>
<parameter name="SupressCheck" value="false"/>
<parameter name="Verbosity"> MESSAGE </parameter>
<parameter name="RandomSeed" value="1234567890" />
</global>
```

## Define processor types and parameters

```
<processor name="FTDPixelPlanarDigiProcessor" type="DDPlanarDigiProcessor">
<parameter name="ForceHitsOntoSurface" type="bool">true </parameter>
<parameter name="SubDetectorName" type="string"> FTD </parameter>
<parameter name="IsStrip" type="bool">false </parameter>
<parameter name="ResolutionU" type="float">0.003 </parameter>
<parameter name="ResolutionV" type="float">0.003 </parameter>
<parameter name="SimTrackHitCollectionName" type="string" lcioInType="SimTrackerHit"> FTD_PIXELCollection </parameter>
<parameter name="TrackerHitCollectionName" type="string" lcioOutType="TrackerHitPlane">FTDPixelTrackerHits </parameter>
</processor>
```

## Run the reconstruction (parameters can be overwritten on the command line)

```
> Marlin MarlinStdReco.xml --global.LCIOInputFiles=bbudsc_3evt_SIM.slcio
```



---

# iLCSoft

- Almost all software now maintained on github
  - <https://github.com/iLCSoft>
  
- iLCSoft v02-01 reference installations available at:
  - `/afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_sl6/v02-01`
  - `/afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_centos7/v02-01`
  - `/cvmfs/ilc.desy.de/sw/x86_64_gcc82_sl6/v02-01`

# Getting started

- Initialize a given release via:

`./afs/desy.de/project/ilcsoft/sw/x86_64_gcc82_sl6/v02-01/init_ilcsoft.sh`

`./cvmfs/ilc.desy.de/sw/x86_64_gcc82_sl6/v02-01/init_ilcsoft.sh`

- Defines a number of global variables, e.g.

- `$ILCSOFT`, `$DD4hep_DIR`, `$LCIO`, `$lcgeo_DIR`

- Can now call all the iLCSoft binaries

- `ddsim -h`

- `Marlin -h`

- `dumpevent -h`

- Standard ILD configuration is available online at

<https://github.com/iLCSoft/ILDConfig/tree/master/StandardConfig/production>

# Getting Involved

- Contact the ILD or SiD groups
- Physics Analyses
  - Would be easiest for individuals just getting started in ILC physics analyses to use the existing full simulation miniDSTs and apply the analysis that they have developed using the fast simulations.
  - Investigate sources of discrepancies (extra “fake” tracks, missing tracks, etc.) using the `lcio::RelationNavigator` to plot some *truth* vs. *reconstructed* quantities.
- Detector Design
  - Is your favorite detector technology available in the simulation package? If not, implement it. If so, tweak some of the readout parameters.
- Software Contributions
  - Read the documentation, examine the source code.
  - Is your favorite algorithm available in the sim/recon chain?
    - If so, can it be improved. If not, implement it!
  - Run a profiler, improve the speed or memory footprint.
- When ready to generate your own events, start small.
  - Use the internal particle gun to generate “single particle” events.
  - Study efficiency/resolution as a function of polar angle, etc.
- Bootstrap from there to simple resonances, single jets, etc.

# Additional Information

- LCIO - <http://lcio.desy.de>

*A persistency framework for linear collider simulation studies (CHEP 2003)*

<https://arxiv.org/pdf/physics/0306114.pdf>

- DD4hep - <http://aidasoft.web.cern.ch/DD4hep>

*DD4hep: A Detector Description Toolkit for High Energy Physics Experiments (CHEP 2014)*

<http://cds.cern.ch/record/1670270/files/AIDA-CONF-2014-004.pdf>

- DDG4

*DDG4 A Simulation Framework based on the DD4hep Detector Description Toolkit (CHEP 2015)*

<http://cds.cern.ch/record/2134621/files/pdf.pdf>