

Intelligent Detector Design

Norman Graf (SLAC)

(for the ALCPG Simulation & Reconstruction WG)

TIPP, Chicago

June 11, 2011

LCD Simulation Mission Statement

- Provide full simulation capabilities for Lepton Collider physics program:
 - Physics simulations
 - Detector designs
 - Reconstruction and analysis
- Need flexibility for:
 - New detector geometries/technologies
 - Different reconstruction algorithms
 - Different machine environments
- Limited resources demand efficient solutions, focused effort.

Overview: Goals

- Facilitate contribution from physicists in different locations with various amounts of time available.
- Use standard data formats, when possible.
- Provide a general-purpose framework for physics software development.
- Develop a suite of reconstruction and analysis algorithms and sample codes.
- Simulate benchmark physics processes on different full detector designs.

Detector Performance Studies

- The ILC community recently finished a very intensive round of detector performance and optimization studies, culminating in the submission of LOI's and is engaged in preparing more detailed updates for the DBD in 2012.
- The CLIC community is currently engaged in an aggressive effort to provide a CDR in 2011.
- Muon Collider physics and detector community beginning to study the experimental environment.
- Number of smaller experiments using various components of this toolkit.

LCIO



ALCPG

SiD

ECFA-LC

LDC

ACFA-LC

GLD

slic

org.lcsim

Java

MOKKA

MarlinReco

C++

JUPITER

Satellites

root

LCIO



ALCPG

SiD

ECFA-LC

LDC

ACFA-LC

GLD

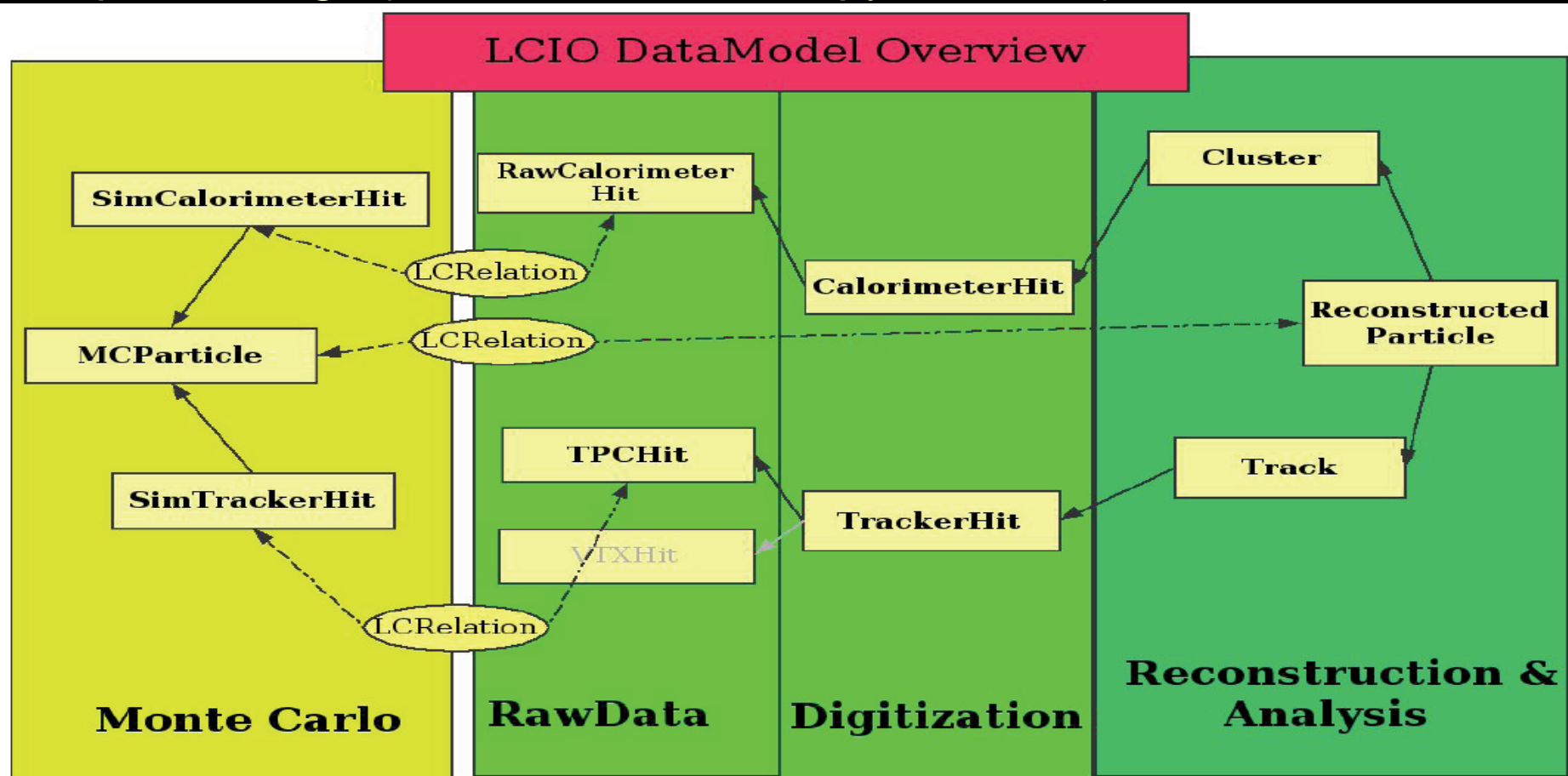
LCIO

Common Data Model

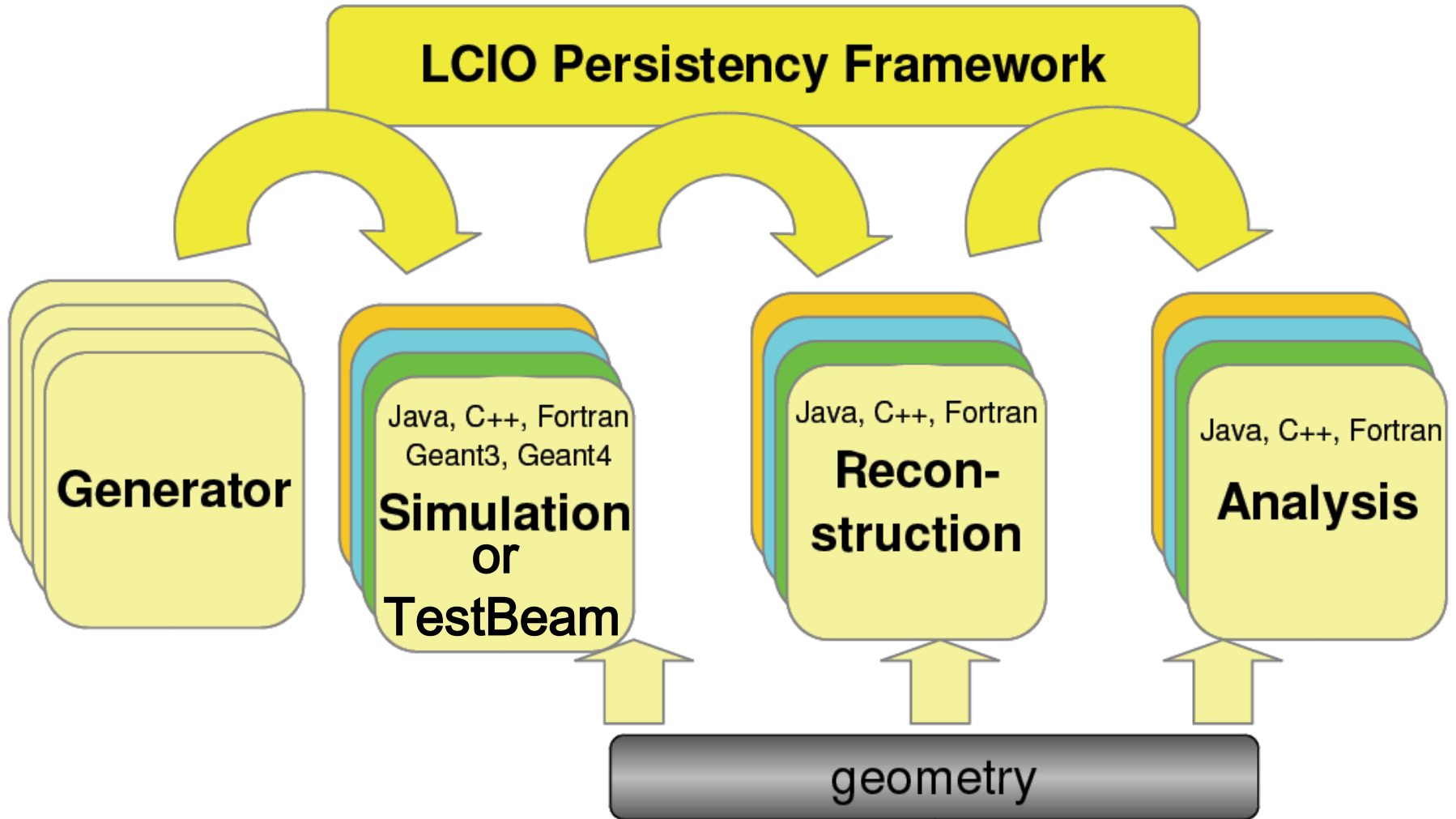
Common IO Format

LCIO Overview

- Object model and persistency format for HEP events
 - MC simulation
 - Test Beam data
 - Reconstructed Objects
- Multiple bindings (C++, Java, Fortran, python, root)



LCIO Overview



LCIO Interoperability

- All three regional LCD simulators write LCIO
 - Cross-checks between data from different simulators
 - Read/write LCIO from
 - Fast MC / Full Simulation / Test Beams
 - Different detectors
 - Different reconstruction tools
 - Different frameworks, languages, operating systems
- Reconstruction also targets LCIO
 - Can run simulation or reconstruction in one framework, analysis in another.
 - Generate events in Jupiter, analyze in MarlinReco.
 - org.lcsim to find tracks in Java, LCFI flavor-tagging to find vertices using MarlinReco (C++) package.

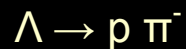
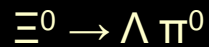
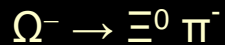
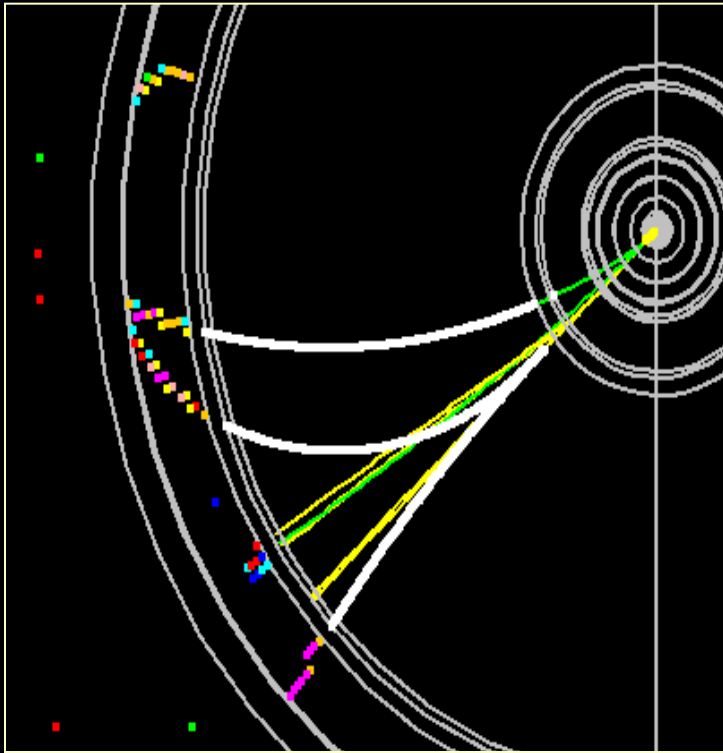
Fast Detector Response Simulation

- Covariantly smear tracks with matrices derived from geometry, materials and point resolution using Billoir's formulation.
 - Analytically from geometry description.
- Smear neutrals according to expected calorimeter resolution (EM for γ , HAD for neutral hadrons)
 - Derived from full Geant4 simulations
- Create reconstructed particles from tracks and clusters (γ , e , μ from MC, $\pi^{+/-}$, K_L^0 for others)
- Can also dial in arbitrary effective jet energy resolution.
 - Derived from full simulation, reconstruction & analysis.
- “Supersymmetry, the *ILC*, and the LHC *inverse problem*”

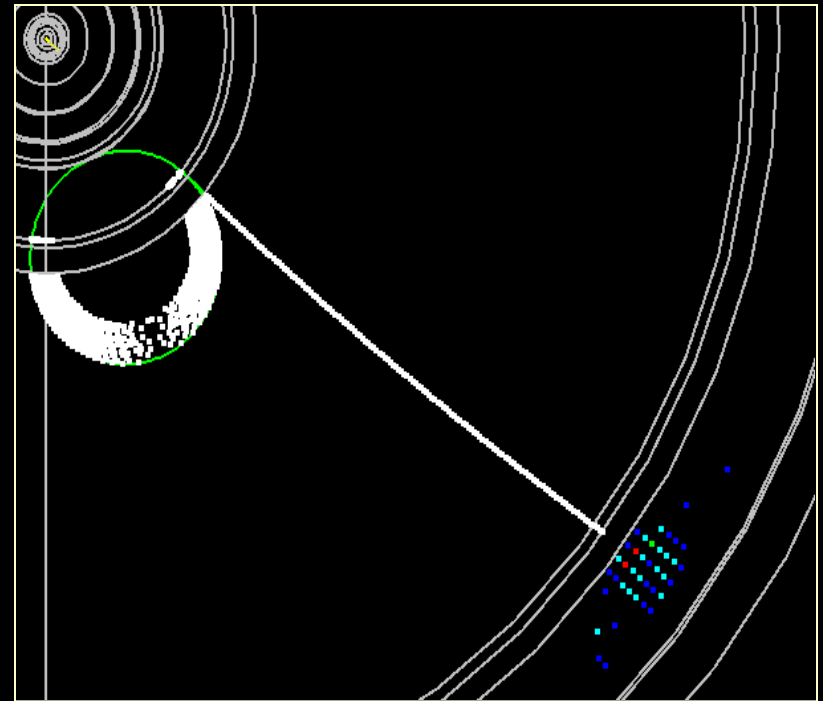
lelaps

- Fast detector response package.
- Handles decays in flight, multiple scattering and energy loss in trackers.
- Parameterizes particle showers in calorimeters.
- Produces detector data at the hit level.
 - Feeds directly into full reconstruction.
- Uses runtime geometry (compact.xml → godl).

Lelaps: Decays, dE/dx , MCS



$\pi^0 \rightarrow \gamma \gamma$ as
simulated by Lelaps for the
LDC model.



gamma conversion as
simulated by Lelaps for the
LDC model.

Note energy loss of electron.

Detector Design (GEANT 4)

- Need to be able to flexibly, but believably simulate the detector response for various designs.
- The daunting machine backgrounds expected at the Muon Collider will require detailed full detector simulations.
- GEANT is the de facto standard for HEP physics simulations.
- Use runtime configurable detector geometries
- Write out “generic” hits to digitize later.

Full Detector Response Simulation

- Use Geant4 toolkit to describe interaction of particles with matter and fields.
- Thin layer of C++ code provides access to:
 - Event Generator input (binary stdhep format)
 - Detector Geometry description (XML)
 - Detector Hits (LCIO)
- Geometries fully described at run-time!
 - In principle, as fully detailed as desired.

Geometry Definition

- Goal was to free the end user from having to write any C++ code or be expert in Geant4 to define the detector.
- All of the detector properties should be definable at runtime with an easy-to-use format.
- Selected xml, and extended the existing GDML format for pure geometry description.

Full Simulation History

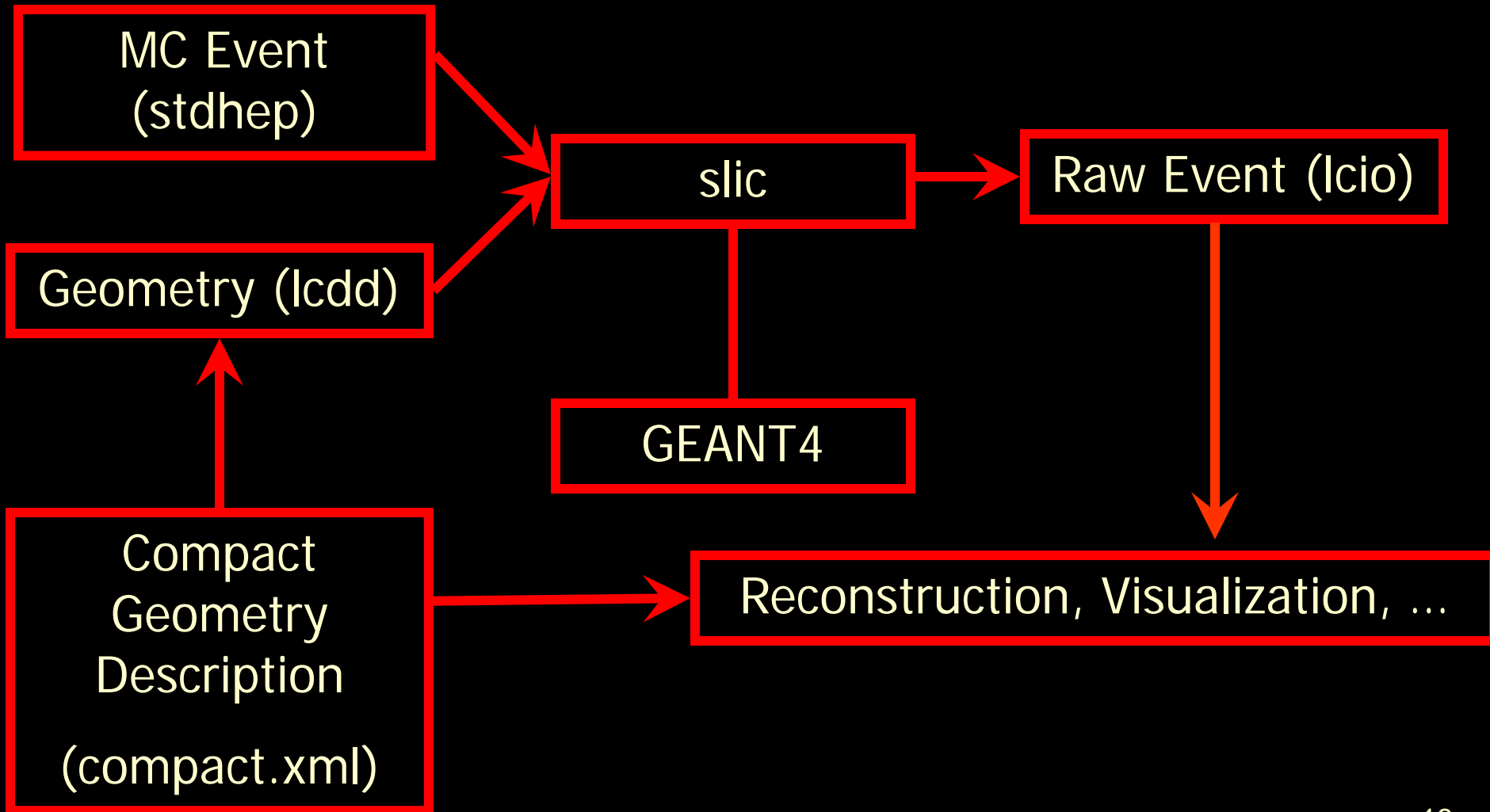
- Provide static binary to run full detector simulations using runtime xml detector descriptions.
 - in-house lcdparm xml format (1998)
 - collaboration with R. Chytrcek on GDML (2000)
- GISMO (C++ GHEISHA + EGS, lcdparm) 1998
- LCDRoot (Geant4 + Root, lcdparm) 1999
- LCDG4 (Geant4 + sio, lcdparm) 2002
- LCS (Geant4 + lcio, lcdparm) 2004
- slic (Geant4 + lcio, GDML) 2005

Fifth Generation



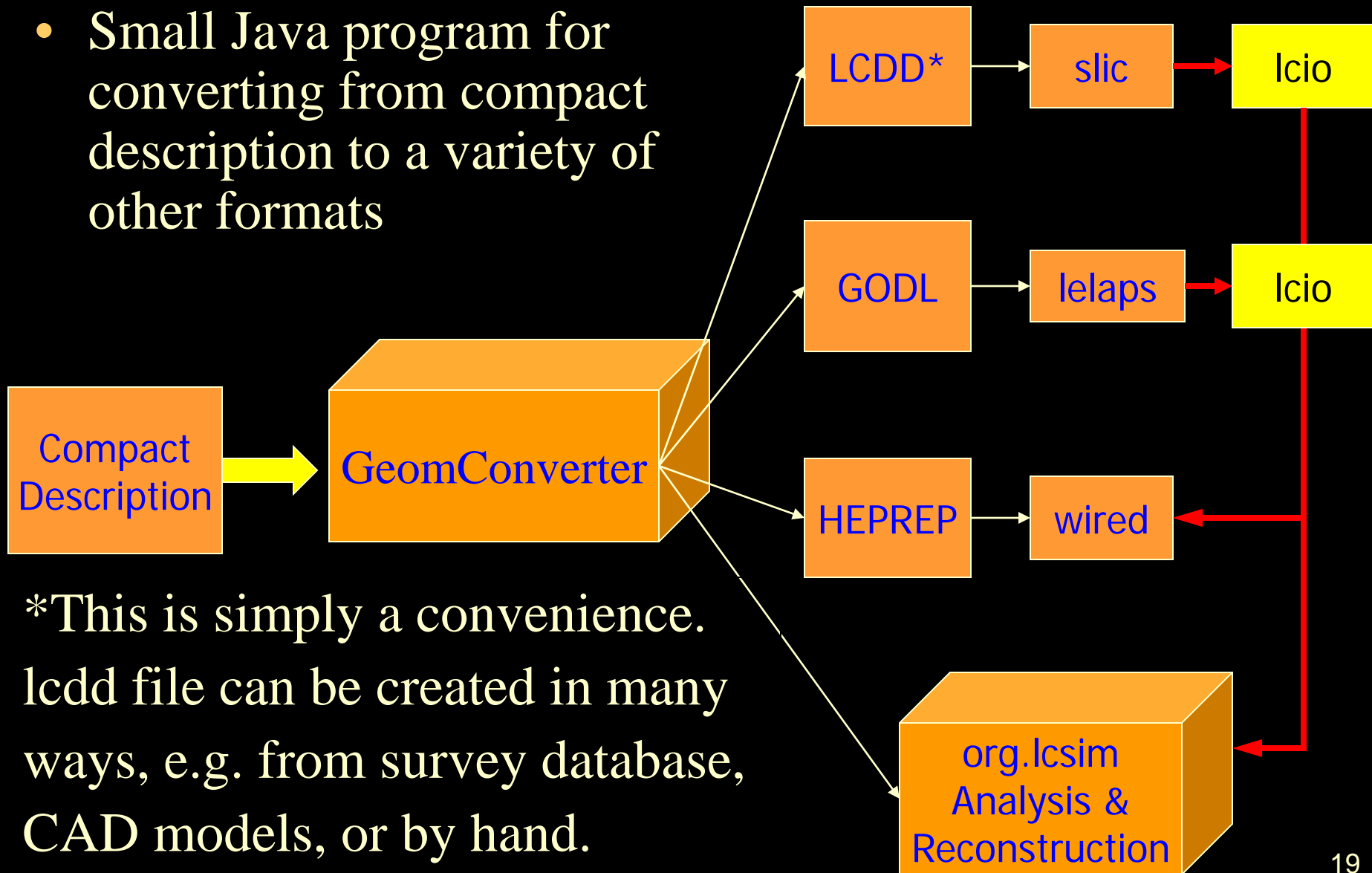
Lockheed-Martin F-22A Raptor
5th Generation Fighter

LC Detector Full Simulation



GeomConverter

- Small Java program for converting from compact description to a variety of other formats



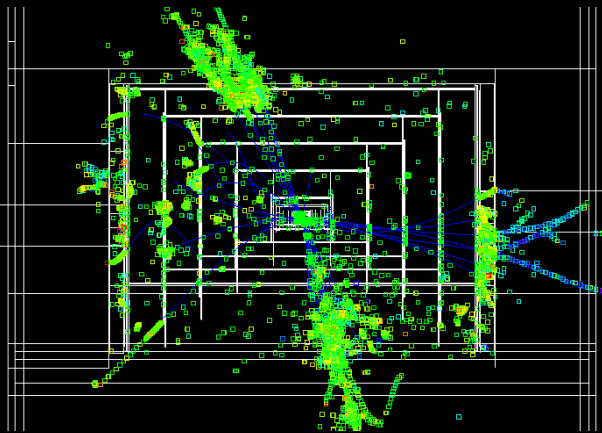
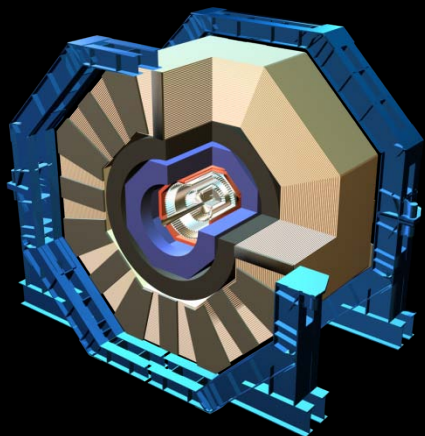
*This is simply a convenience. Icidd file can be created in many ways, e.g. from survey database, CAD models, or by hand.

Detector Variants

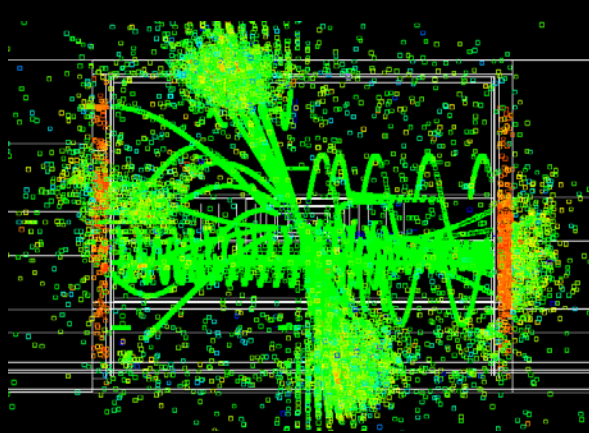
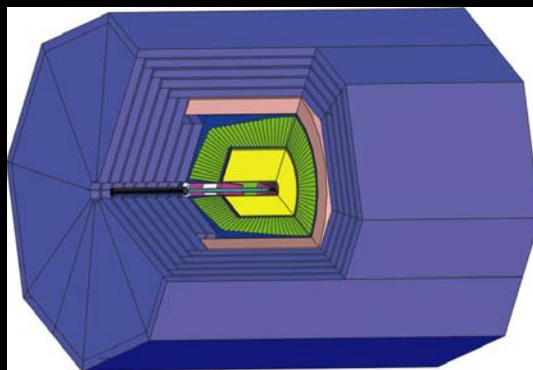
- Runtime XML format allows variations in detector geometries to be easily set up and studied:
 - Absorber materials and readout technologies for sampling calorimeters
 - e.g. Steel, W, Cu, Pb + RPC vs. GEM vs. Scintillator readout
 - Optical processes for dual-readout or crystal calorimeters
 - Layering (radii, number, composition)
 - Readout segmentation (size, projective vs. nonprojective)
 - Tracking detector technologies & topologies
 - TPC, Silicon microstrip, pixels, ...
 - “Wedding Cake” Nested Tracker vs. Barrel + Cap
 - Far forward MDI variants, shielding, field strength, etc.

ILC Full Detector Concepts

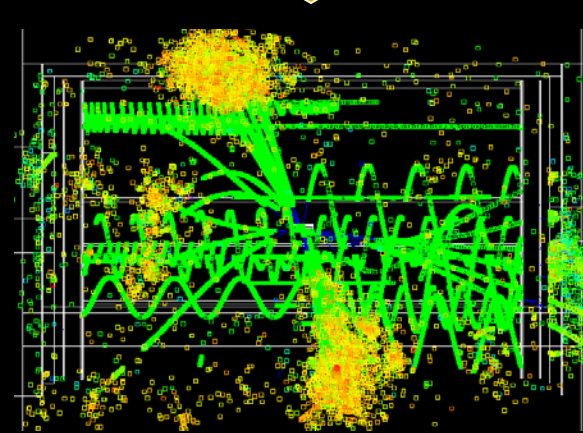
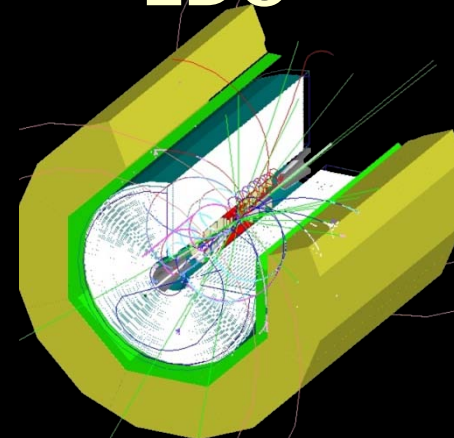
SiD



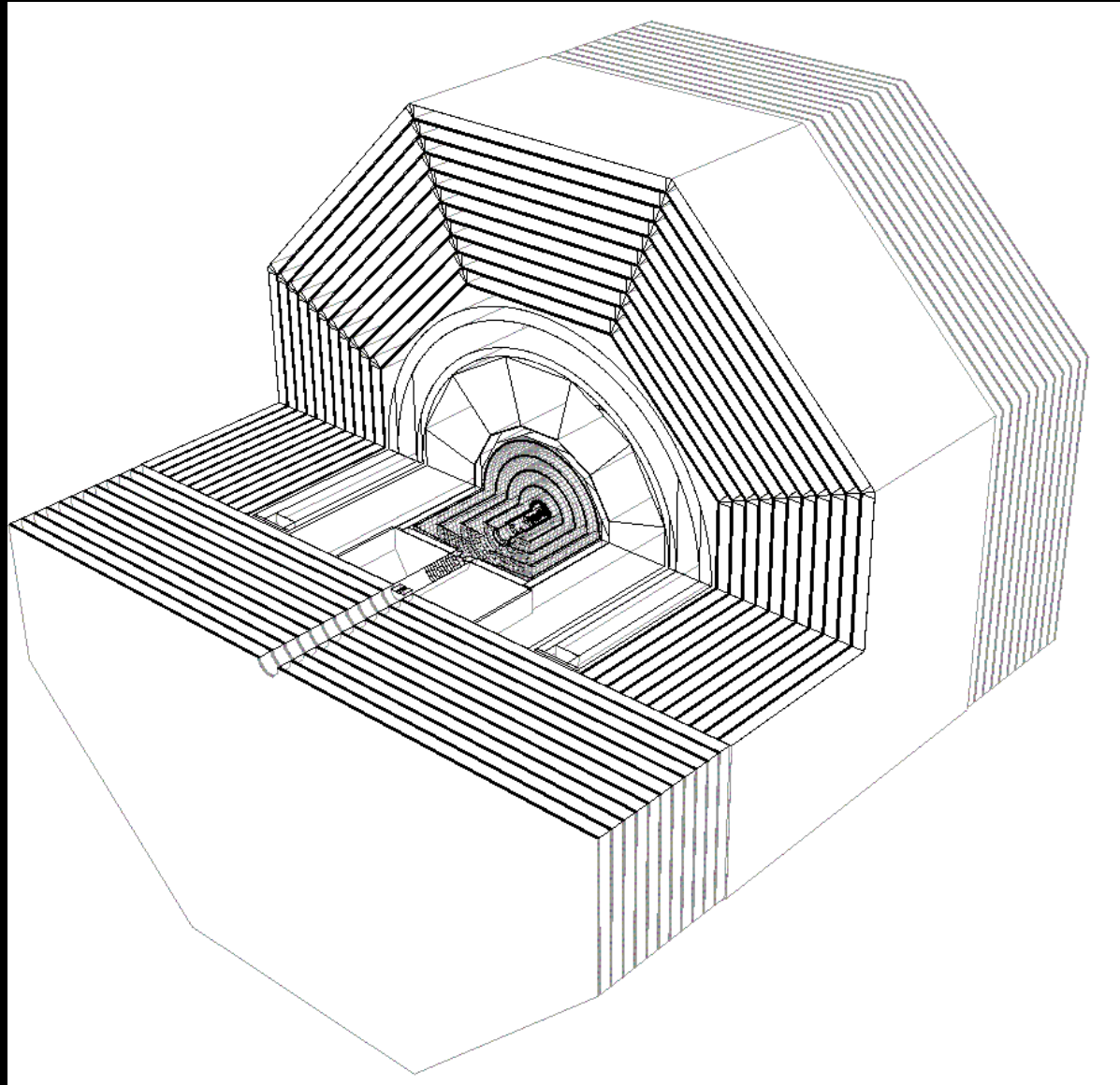
GLD



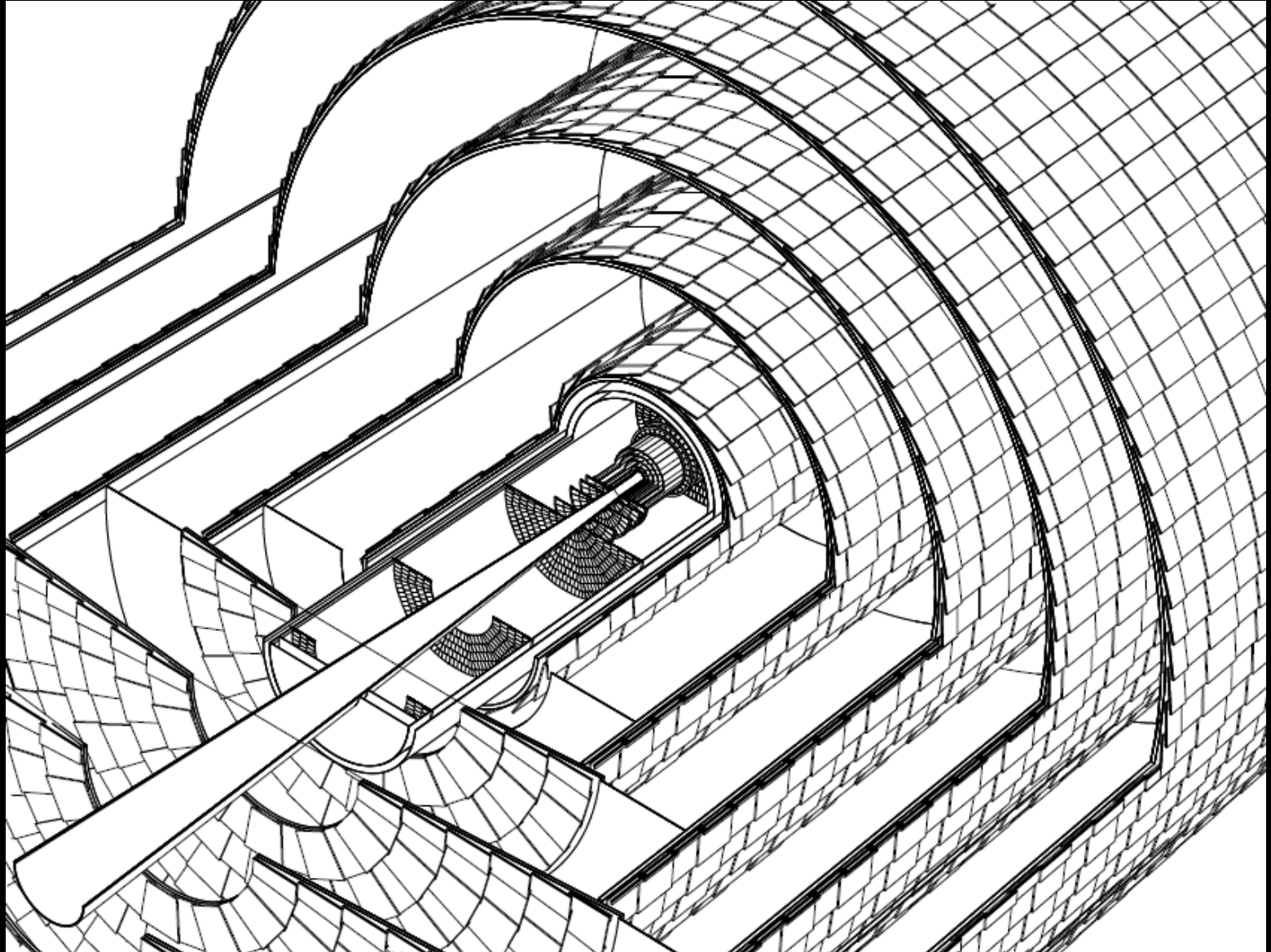
LDC



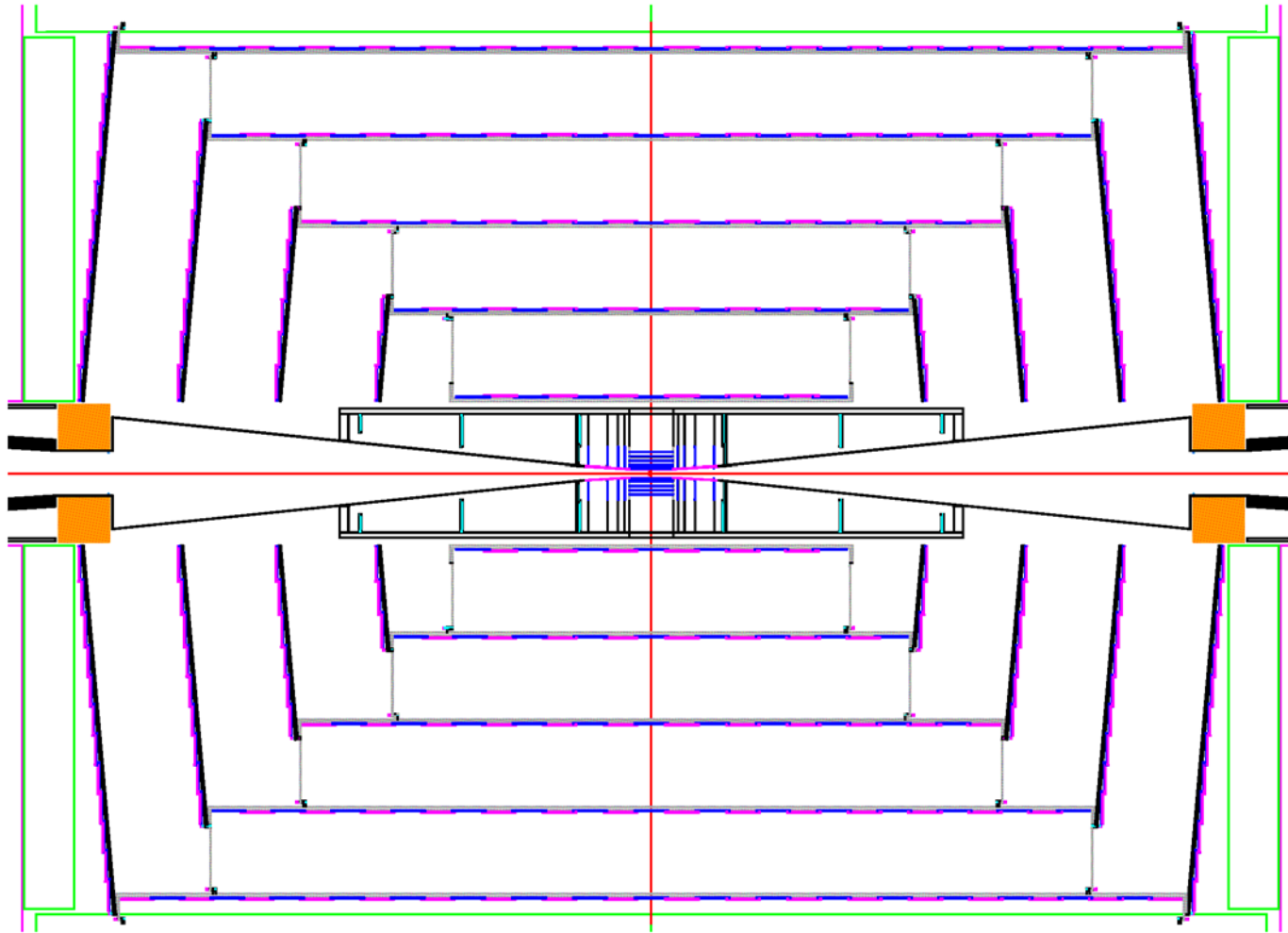
Silicon Detector Concept



Silicon Detector Concept

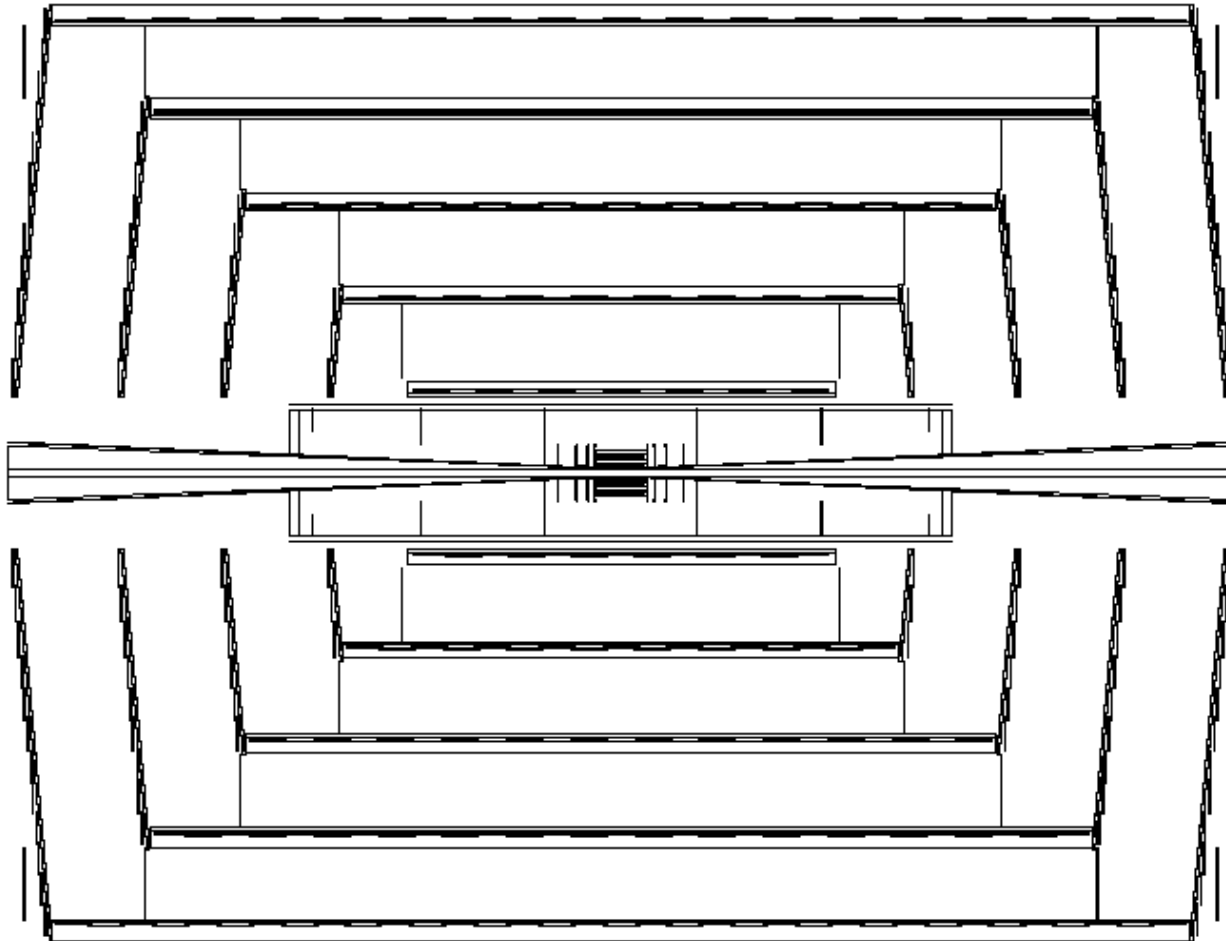


SiD Tracker



CAD Model

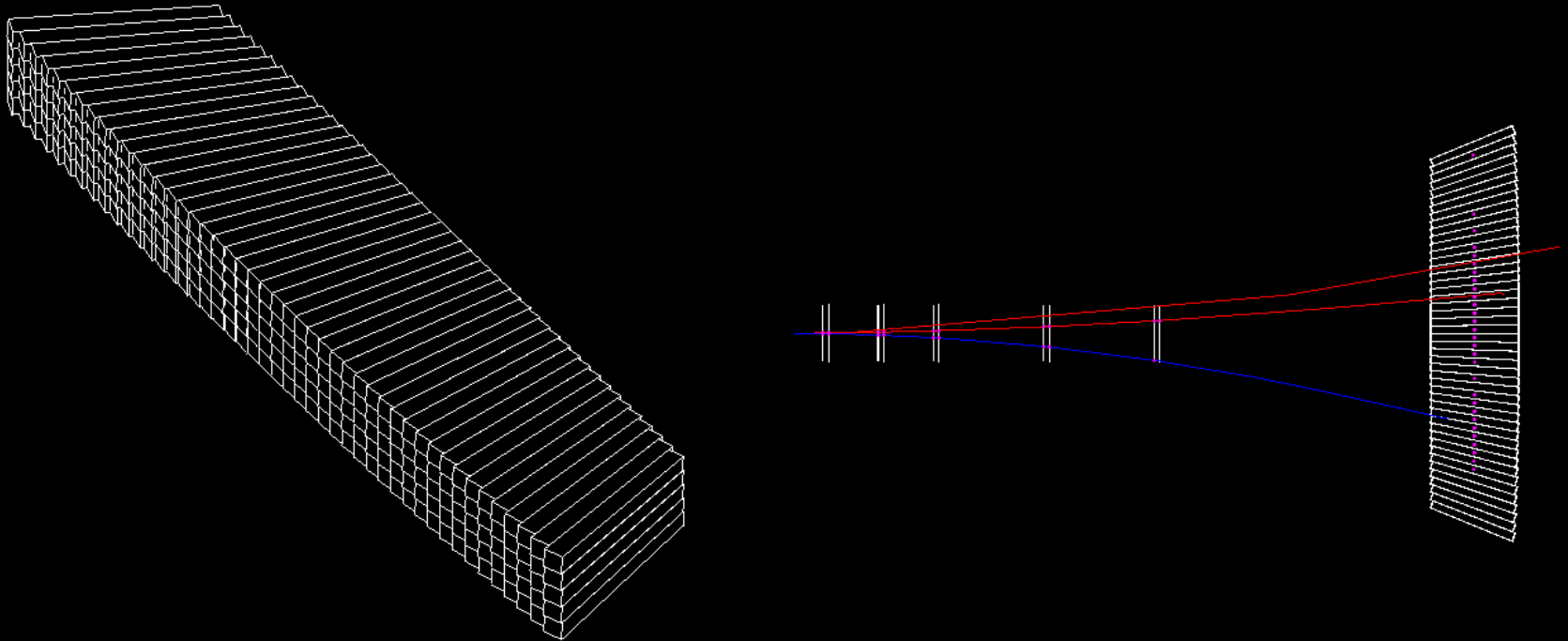
SiD Tracker



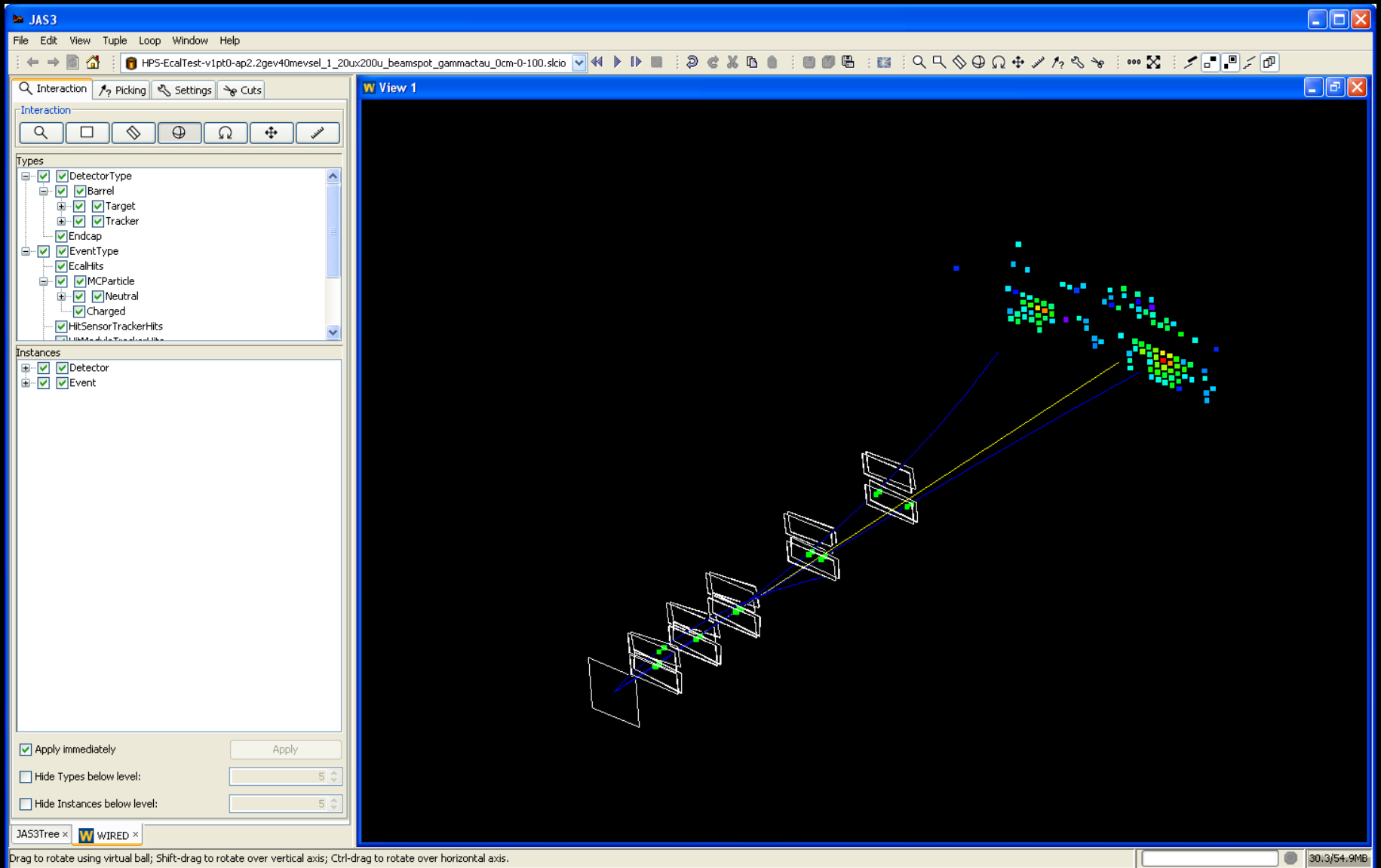
Geant4 Model

Simulating the HPS ECal

- Crystal array geometry and readout is supported in the compact format.



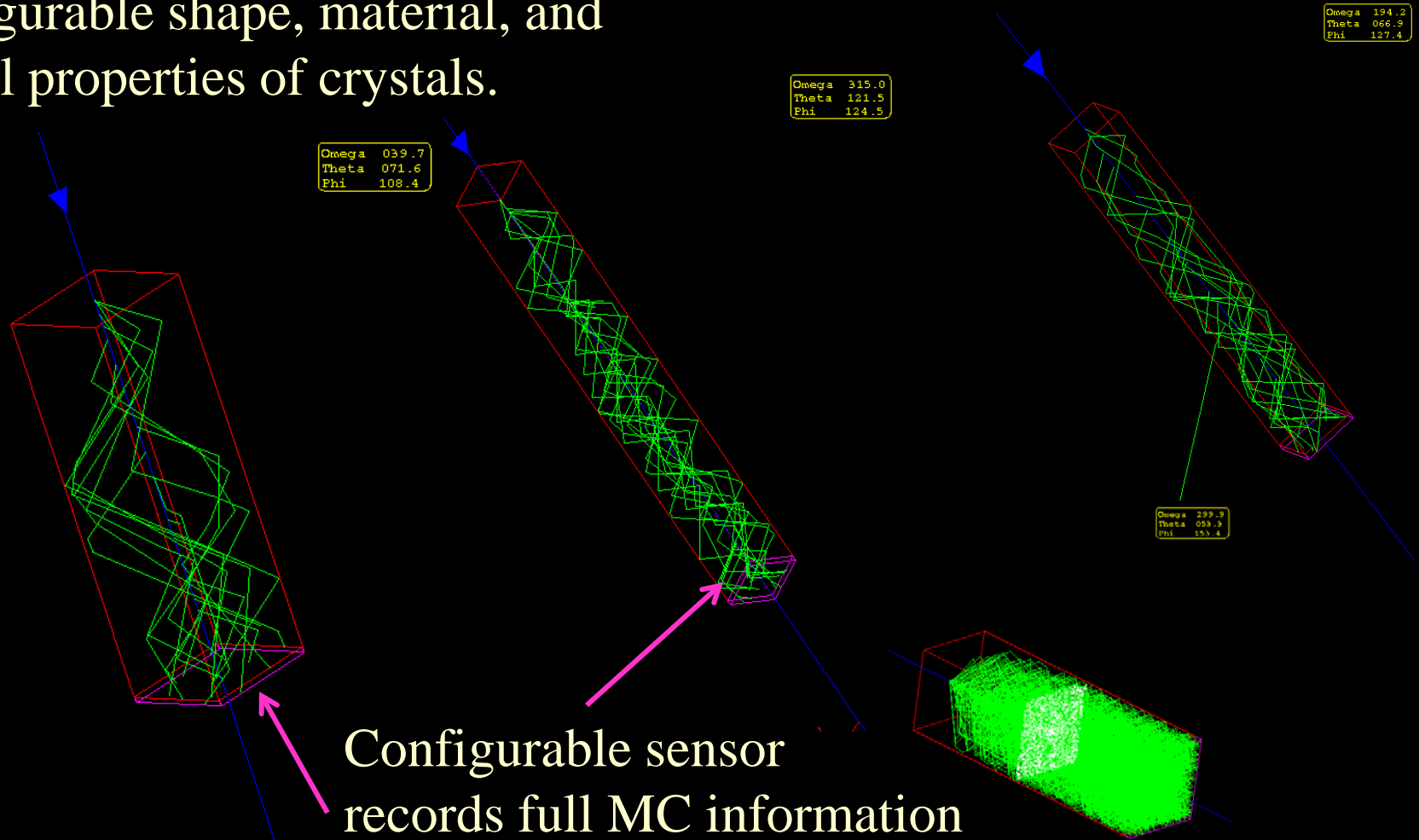
Wired Event Display



Drag to rotate using virtual ball; Shift-drag to rotate over vertical axis; Ctrl-drag to rotate over horizontal axis.

Optical Ray Tracing in Crystals

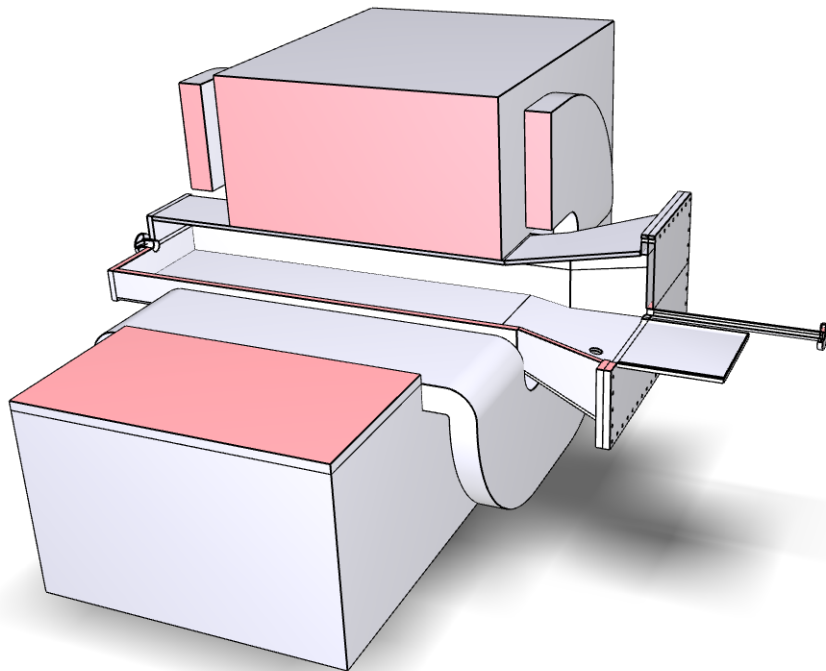
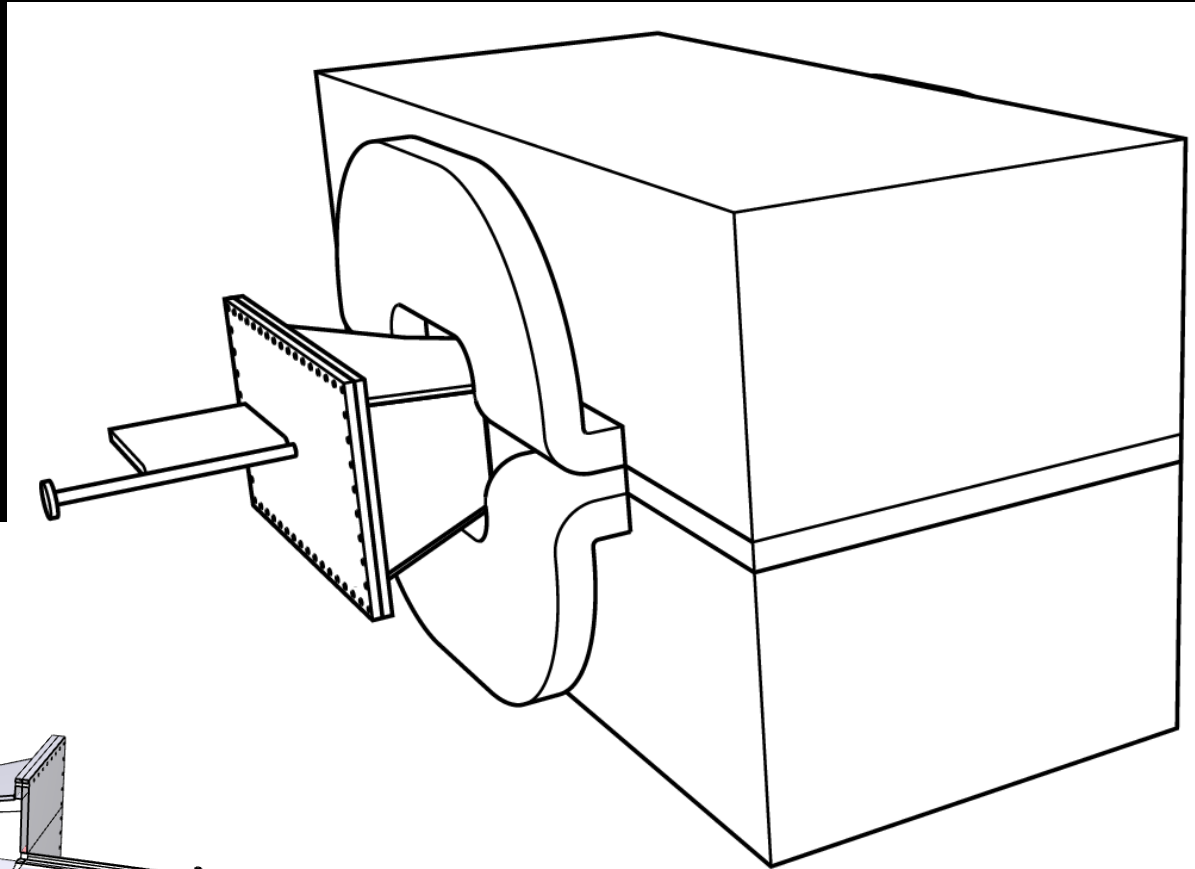
Configurable shape, material, and optical properties of crystals.



Configurable sensor records full MC information about each incident photon

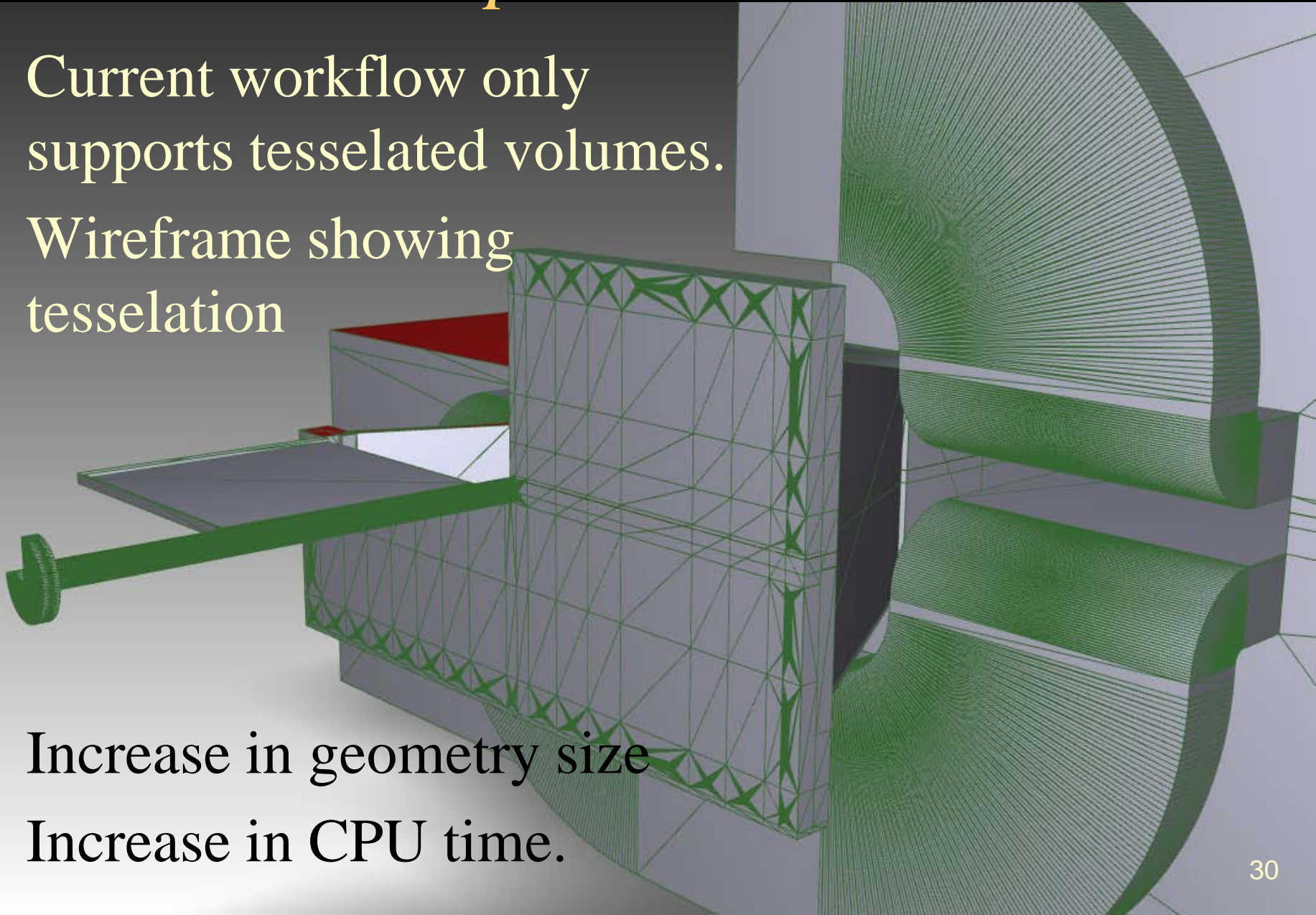
HPS Dipole and Vacuum Vessel

CAD to GDML.
Can be used for
non-sensitive
elements



CAD-imported elements.

- Current workflow only supports tessellated volumes.
- Wireframe showing tessellation



- Increase in geometry size
- Increase in CPU time.

slic: The Executable

- Build static executables on Linux, Windows, Mac.
- Commandline or G4 macro control.
- Only dependence is local detector description file.
 - Trivial Grid usage (no database call-backs, etc.)
 - Grid ready, Condor and lsf scripts available.
- Event input via stdhep, particle gun, ...
- Detector input via GDML, lcdd
- Response output via LCIO using generic hits.

Reconstruction/Analysis Overview

- Java based reconstruction and analysis package
 - Runs standalone or inside Java Analysis Studio (JAS)
 - Fast MC → Smearred tracks and calorimetry clusters
 - Full Event Reconstruction
 - Beam background overlays at detector hit level, including time offsets.
 - detector readout digitization (CCD pixels, Si μ -strips, TPC pad hits)
 - *ab initio* track finding and fitting for ~arbitrary geometries
 - multiple calorimeter clustering algorithms
 - Individual Particle reconstruction (cluster-track association)
 - Analysis Tools (including WIRED event display)
 - Physics Tools (Jet Finding, Vertex Finding, Flavor Tagging)
- Write once run, run anywhere
 - Exact same libraries run on all platforms (Windows, Mac, Linux(es), Grid) using the Java Virtual Machine.

Tracking

- Analytic covariance matrices available for fast MC smearing for each detector.
- Track “cheater” available for studies of full detector simulation events. Assigns hits on basis of MC parentage.
- Ab initio track finding packages.
- Fitting code incorporating multiple scattering and energy loss via weight matrix or Kalman Filter available.

Tracking Detector Readout

- Hits in Trackers record full MC information.
- Module tiling and signal digitization is deferred to analysis stage.
 - Used to rapidly study many possible solutions.
- Fully-featured package to convert MC hits in silicon to pixel hits. Fully configurable at runtime.
MC Hits → Pixel ID & ADC → Clusters → Hits ($x \pm \delta x$)
- Can correctly study occupancies, overlaps, ghost hits, etc.

Track Finding

- Standalone pattern recognition code for 1D (e.g. Si μ strip) and 2D (e.g. Si pixel) hits.
 - High efficiency, even in presence of backgrounds.
 - Efficient at low momentum.
- Conformal-mapping pattern recognition also available, applicable also to TPC.
- MIP stubs in highly segmented calorimeters also provide track candidates, propagate inwards to pick up tracker hits.

Java Analysis Studio (JAS)

- Integrated Development Environment (editor, compiler)
- Cross-platform physics analysis environment with iterative, event-based analysis model
 - quick development, debugging, ad hoc analysis
 - additional functionality with plugins
- Dynamically load / unload Java analysis drivers
 - Supports distributed computing.
- Plotting and fitting and analysis (cuts, scripting) engine
 - 1D, 2D histograms, clouds, profiles, dynamic scaling, cuts
 - high-quality output to vector or raster formats
- Integrated event browser and event display

JAS editor/compiler

The screenshot displays the JAS3 software interface. The main window is titled "JAS3" and contains a menu bar (File, Edit, View, Tuple, Loop, LCIO, Window, Help) and a toolbar. The left sidebar shows a file tree with "DataSets" and "Programs" folders, and a sub-folder "CombinedConeClusterAnalysisDriver". The central pane shows the source code for "CombinedConeClusterAnalysisDriver.java". The code includes a class definition and a process method that iterates over calorimeter hits to form clusters. The right pane shows a visualization of the data, with a grid and several clusters of points (green and yellow) connected by white lines. The status bar at the bottom indicates "9:23:33 PM ----- compile successful" and "Analyzed 1 records in 190ms".

```
70 CombinedConeClusterAnalysisDriver.java
71 *
72 * Created on March 3, 2006, 4:21 PM
73 *
74 * To change this template, choose Tools | Template Manager
75 * and open the template in the editor.
76 */
77
78 /**
79 *
80 * @author ngraf
81 */
82 public class CombinedConeClusterAnalysisDriver extends Driver
83 {
84     private FixedConeClusterer _clusterer;
85     /** Creates a new instance of CombinedConeClusterAnalysisDriver */
86     public CombinedConeClusterAnalysisDriver()
87     {
88         double radius = 1.2;
89         double seedEnergy = 0.0;
90         double minEnergy = 0.0;
91         _clusterer = new FixedConeClusterer(radius, seedEnergy, minEnergy);
92     }
93
94     protected void process(EventHeader event)
95     {
96         // the list of hit cells to cluster
97         List<CalorimeterHit> cellsToCluster = new ArrayList<CalorimeterHit>();
98         // get all the calorimeter hits in this event...
99         List<List<CalorimeterHit>> collections = event.get(CalorimeterHit.class);
100
101         for (List<CalorimeterHit> collection : collections)
102         {
103             LCMetaData meta = event.getMetaData(collection);
104             System.out.println(meta.getName()+" has "+collection.size()+" hits");
105             for (CalorimeterHit hit : collection)
106             {
107                 // should apply cut here...
108                 // punt for now and add ALL hits
109                 cellsToCluster.add(hit);
110             }
111         }
112
113         System.out.println("Event has "+cellsToCluster.size()+" hit cells");
114     }
115 }
```

JAS3Tree x WIRED x

9:23:33 PM ----- compile successful

Compiler x Record Loop x

Analyzed 1 records in 190ms

24.6/34.4MB

JAS event browser

JAS3

File Edit View Tuple Loop LCIO Window Help

pythia2Polebbar-0-1000_SLIC_v1r13p3_sid00.slcio

DataSets
Programs
ClusterFinding
aida22594aida

e-(E=45.500 status=Documentation)
 e-(E=45.500 status=Documentation)
 e-(E=45.500 status=Documentation)
 Zo(E=91.000 status=Documentation)
 b(E=45.500 status=Documentation)
 b(E=38.330 status=Intermediate)
 unknown(E=91.000 status=Intermediate)
 B*(E=31.648 status=Intermediate)
 pi0(E=2.1998 status=Intermediate)
 pi+(E=4.0914 status=Final State)
 rho0(E=1.2687 status=Intermediate)
 K(892)*o(E=2.8169 status=Intermediate)
 K-(E=2.6171 status=Final State)
 pi0(E=1.5627 status=Intermediate)
 pi0(E=60782 status=Intermediate)
 Delta+(E=3.5210 status=Intermediate)
 Ko_bar(E=2.0863 status=Intermediate)
 Xi0_bar(E=7.6546 status=Intermediate)
 B(s)*o(E=30.926 status=Intermediate)
 gluon(E=7.0819 status=Intermediate)
 gluon(E=91563 status=Intermediate)
 gluon(E=49304 status=Intermediate)

LCSim Event
Run:0 Event: 179

LCIO Event Header

Run	0
Event	179
Time Stamp	Thu Feb 16 10:34:33 PST 2006
Detector Name	sid00

Collections

Name	Type	Size
EcalEndcapHitsNNClusters	org.lcsim.event.Cluster	22
EcalEndcapHitsNNClusters	org.lcsim.event.Cluster	22
EcalBarrHitsNNClusters	org.lcsim.event.Cluster	24
HcalBarrHitsNNClusters	org.lcsim.event.Cluster	7
HcalEndcapHitsNNClusters	org.lcsim.event.Cluster	3
VbEndcapHits	org.lcsim.event.SimTrackerHit	72
EcalBarrHits	org.lcsim.event.SimCalorime...	1036
EcalEndcapHits	org.lcsim.event.SimCalorime...	1084
ForwardEcalEndcapHits	org.lcsim.event.SimCalorime...	33
HcalBarrHits	org.lcsim.event.SimCalorime...	197
HcalEndcapHits	org.lcsim.event.SimCalorime...	129
LuminosityMonitorHits	org.lcsim.event.SimCalorime...	0
MuonBarrHits	org.lcsim.event.SimCalorime...	1

LCSim Event
Run:0 Event: 179

Collection: EcalEndcapHits size:1084 flags:e0000000

id: system	id: layer	id: barrel	id: x	id: y	raw energy (...)	corrected e...	x (mm)	y (mm)	z (mm)	time (ns)
6	0	2	77	304	1.0683E-4	.0084284	271.25	1065.8	-1683.3	6.7236
6	1	2	77	304	1.1771E-4	.0092868	271.25	1065.8	-1687.1	6.7386
6	2	2	77	305	1.0897E-4	.0085971	271.25	1069.2	-1690.9	6.7536
6	3	2	77	306	1.2685E-4	.010008	271.25	1072.8	-1694.6	6.7685
6	4	2	77	306	4.6335E-5	.0036556	271.25	1072.8	-1698.3	6.7834
6	4	2	77	307	1.2153E-4	.0095883	271.25	1076.2	-1698.3	6.7840
6	10	2	83	303	3.6268E-4	.028614	292.25	1062.2	-1720.8	7.2580
6	26	2	105	200	2.0871E-5	.0016466	369.25	701.75	-1798.3	101.67
6	21	2	55	297	1.4128E-5	.0011147	194.25	1041.2	-1767.1	12.514
6	9	2	77	309	8.8875E-4	.070119	271.25	1083.2	-1717.1	6.8521
6	6	2	73	299	3.4643E-4	.027332	257.25	1048.2	-1705.8	7.4273
6	7	2	93	313	6.7790E-4	.053483	327.25	1097.2	-1709.6	8.2976

JAS3Tree x WIRED x

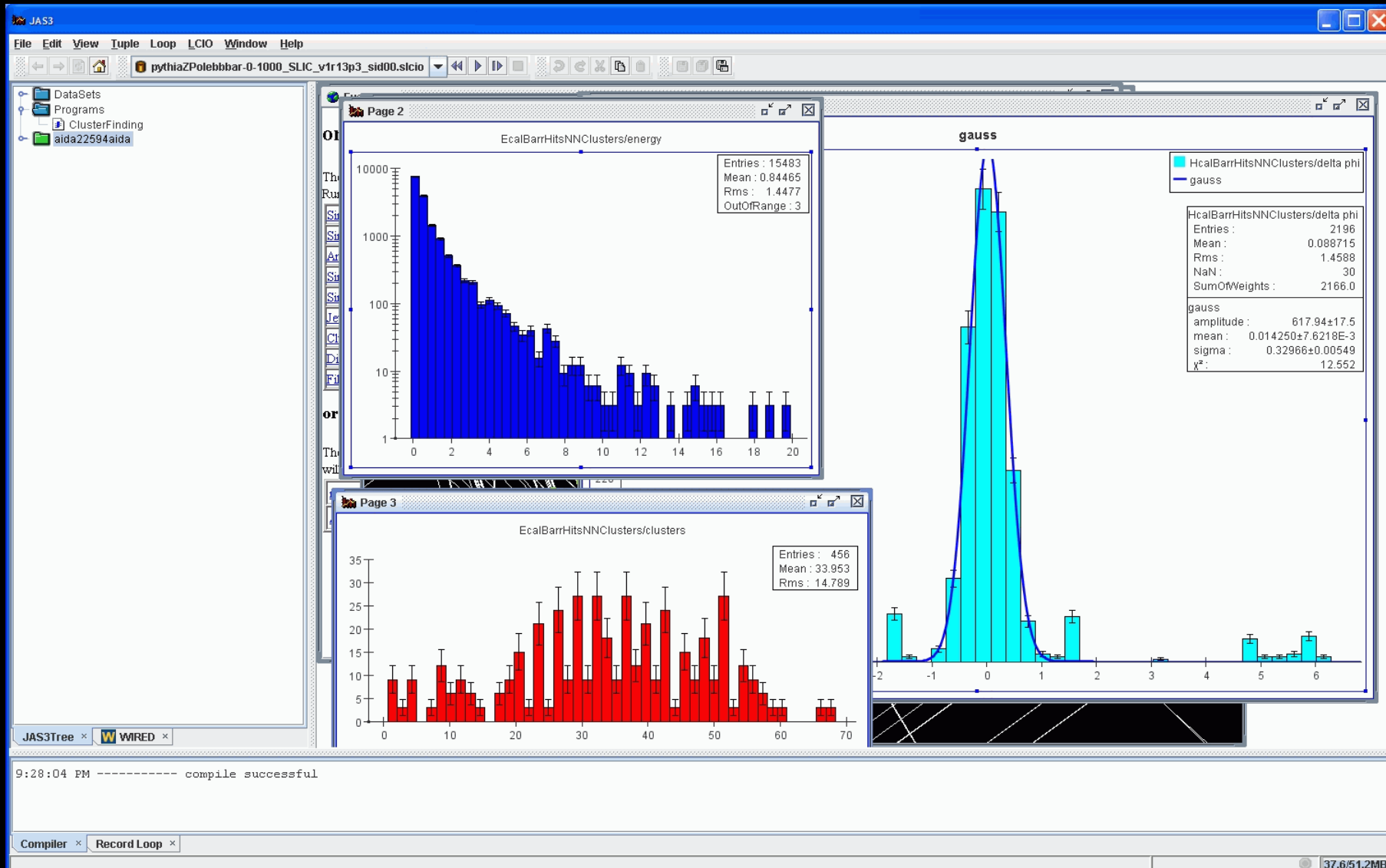
9:28:04 PM ----- compile successful

Compiler x Record Loop x

Analyzed 151 records in 114485ms

38.4/51.2MB

JAS histogramming/fitting



Wired LCD Event Display

The screenshot shows the JAS3 software interface. The main window displays a complex event visualization with particle tracks and hits. The left sidebar contains a list of detector types and hit types, and the right sidebar shows event statistics and a histogram.

Interaction List:

- DetectorType
- EventType
- EcalEndcapHitsNNClusters
- VtxEndcapHits
- EcalBarrHits
- EcalEndcapHits
- ForwardEcalEndcapHits
- HcalBarrHits
- HcalEndcapHits
- LuminosityMonitorHits
- MuonBarrHits
- MuonEndcapHits
- TkrBarrHits
- TkrEndcapHits
- VtxBarrHits
- EcalBarrHitsNNClusters
- HcalEndcapHitsNNClusters
- HcalBarrHitsNNClusters
- MCPParticle
 - Neutral
 - Charged

Event Statistics:

sNNClusters/delta phi	2196
sNNClusters/delta phi	0.088715
	1.4588
ghts :	30
	2166.0
	617.94±17.5
	0.014250±7.6218E-3
	0.32966±0.00549
	12.552

Histogram:

Bin	Count
5	~10
6	~15

9:28:04 PM ----- compile successful

Compiler x Record Loop x

37.8/51.2MB

Validated

- This suite of software tools provides:
 - Physics event generation & bindings to most legacy generators through the stdhep format.
 - Full detector response simulation using precompiled binaries & runtime geometry definition (no coding!).
 - Full detector digitization (x-talk, noise, diffusion, etc.)
 - Hit-level overlay of arbitrary background events.
 - Access to other LCIO-compliant software frameworks.
 - Full ab-initio event reconstruction and analysis suites.
 - Tested on hundreds of millions of events.
- “From zero to analysis in 15 minutes.”

User base

- ILC physics and detector community
 - primarily US and UK members of SiD
- CLIC physics and detector community
 - CERN-based SiD' studies
- Muon collider physics and detector community
- JLAB heavy-photon search proposals
 - HPS: SLAC-based, fixed target, forward detector
 - DarkLight: MIT-based, gas-jet, asymmetric detector.
- FNAL dual-readout crystal calorimetry

Simulation Summary

- ALCPG sim/reco supports an ambitious international detector simulation effort. Goal is flexibility and interoperability.
- Provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description.
- Reconstruction & analysis framework was used to characterize the Silicon Detector and was essential to that concept's successful validation in the LOI process.
- LCIO provides interoperability with tools developed in other regions (e.g. jet flavor tagging (LCFI), particle flow (Pandora)), other languages (FORTRAN, java, C++, python) and other analysis frameworks (e.g. Marlin, root).

Additional Information

- Wiki - <http://confluence.slac.stanford.edu/display/ilc/Home>
- lcsim.org - <http://www.lcsim.org>
- ILC Forum - <http://forum.linearcollider.org>
- LCIO - <http://lcio.desy.de>
- SLIC - <http://www.lcsim.org/software/slic>
- LCDD - <http://www.lcsim.org/software/lcdd>
- JAS3 - <http://jas.freehep.org/jas3>
- AIDA - <http://aida.freehep.org>
- WIRED - <http://wired.freehep.org>