ALCPG Simulation/Reconstruction

Norman Graf
SLAC
June 27, 2005

Simulation Group's Mission Statement

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

"Standard LC MC Sample"

- Generate an inclusive set of MC events with all SM processes + backgrounds arising from beam- and brems-strahlung photons and machine-related particles.
- Used for realistic physics analyses and used by the ILC physics community to represent a "standard" sample.
- Samples will be generated at several energy points to systematically study different ILC configurations.
  - 1 ab^{-1} @ 350, 500 & 1000 GeV center-of-mass energy
- Defining and generating "benchmark" physics processes which stress the detector capabilities.

WHIZARD ab^{-1} Data Sets 2005

- Testing new features of WHIZARD 1.40:
  - Time limit, Output file size limit, Pt kick
  - Improved phase space treatment for complicated final states such as $e^+e^- \rightarrow e^+e^-e^+e^-e^+e^-$
  - Interface to GuineaPig files (replaces CIRCE)
- Fixing some missing features (radiative returns)
- Sufficient number of Guinea-Pig files are already available at 500GeV.
- Event Generation to start soon.
- Plan to keep files on disk during Snowmass.

"Signal" and Diagnostic Samples

- simple single particles: γ, μ, e, $\pi^{+/-}$, n, …
- composite single particles: π^0, ρ, K^0_S, τ, ψ

ftp://ftp-lcd.slac.stanford.edu/lcd/ILC/singleParticle/stdhep/

- Z Pole events

ftp://ftp-lcd.slac.stanford.edu/lcd/ILC/ZPole/stdhep/

- WW, ZZ, $t\bar{t}$, $q\bar{q}$, $\tau^+\tau^-$, $\mu^+\mu^-$, Zγ, Zh

ftp://ftp-lcd.slac.stanford.edu/lcd/ILC/ILC500

- Also "Benchmark" samples after they have been identified.

Fast Detector Response Simulation

- Covariantly smear tracks with matrices derived from geometry, materials and point resolution using Billoir's formulation.

  http://www.slac.stanford.edu/~**schumm/lcdtrk**

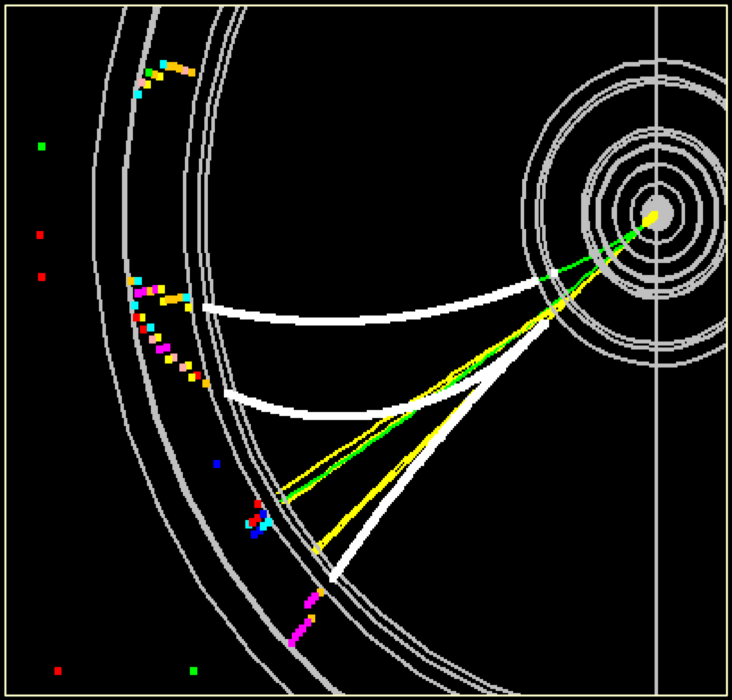
- Smear neutrals according to expected calorimeter resolution (EM for γ, HAD for neutral hadrons)
- Create ReconstructedParticles from tracks and clusters (γ, e, μ from MC, $\pi^{+/-}$, n^0 for others)
- Work just starting to add complexity to process.

lelaps

- Fast detector response package (Willy Langeveld).
- Handles decays in flight, multiple scattering and energy loss in trackers.
- Parameterizes particle showers in calorimeters.
- Targets both sio and lcio at the hit level.
- Recent overhaul of detector description.
  - Was hardcoded, now runtime definable
  - NOT XML, but its own format (godl)
  - possible compact.xml $\rightarrow$ godl

http://lelaps.freehep.org/index.html

Lelaps: Decays, dE/dx, MCS

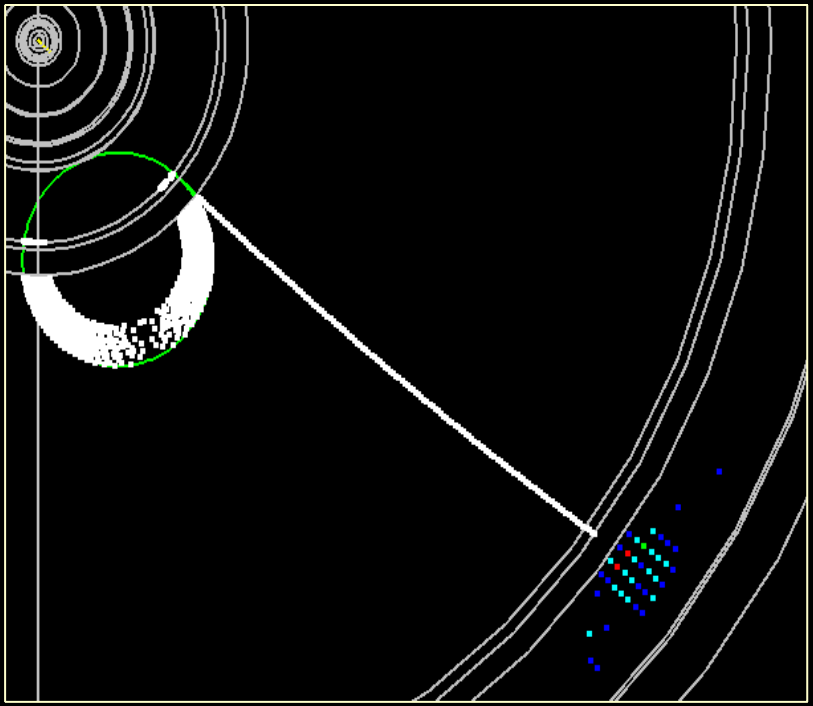


$\Omega^- \rightarrow \Xi^0 \pi^-$

$\Xi^0 \rightarrow \Lambda \pi^0$

$\Lambda \rightarrow p \pi^-$

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

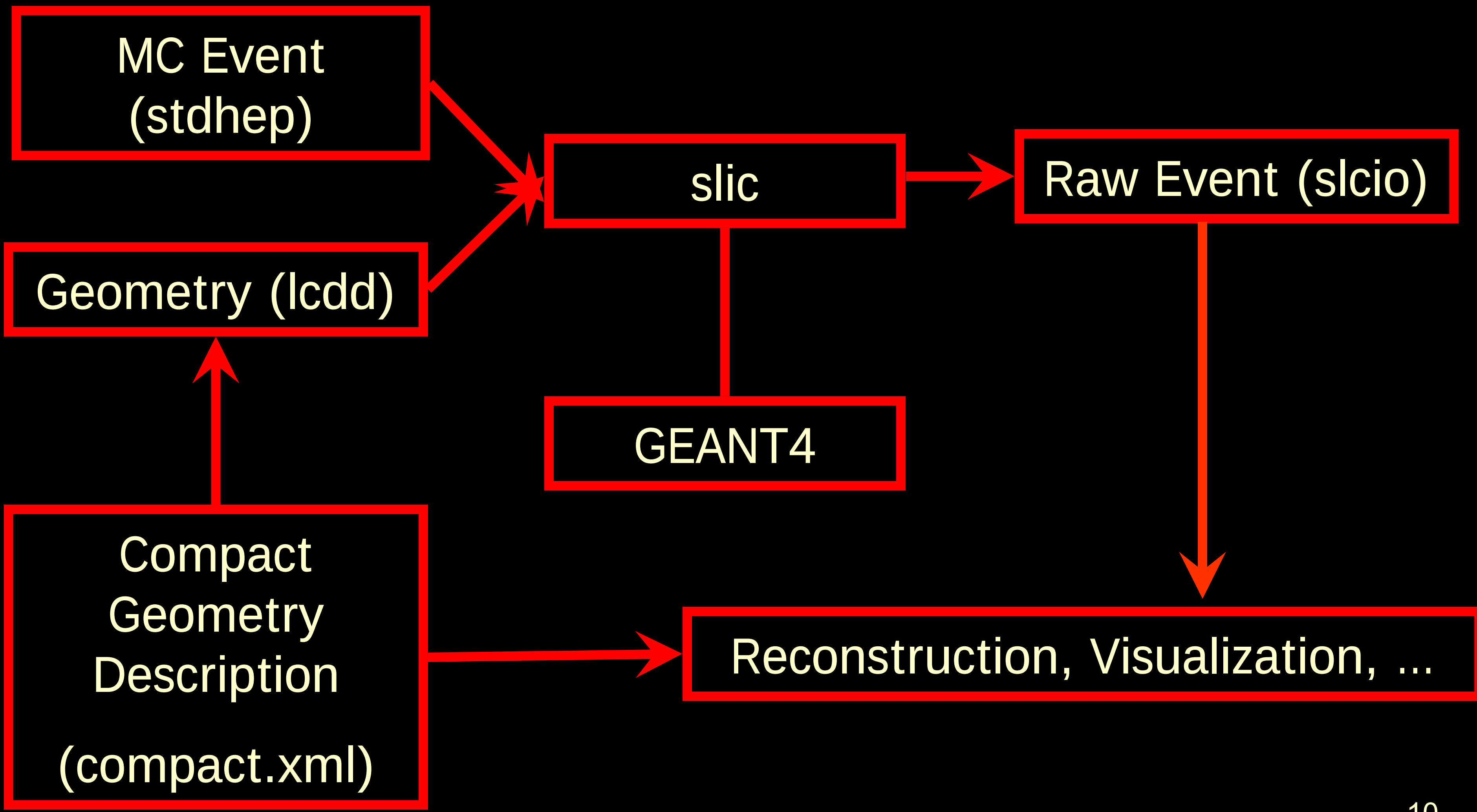


gamma conversion as simulated by Lelaps for the LCD LD model.

Full Detector Response Simulation

- Use Geant4 toolkit to describe interaction of particles with matter.
- Thin layer of LC-specific C++ 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.
  - In practice, will explore detector variations with simplified approximations.

LC Detector Full Simulation

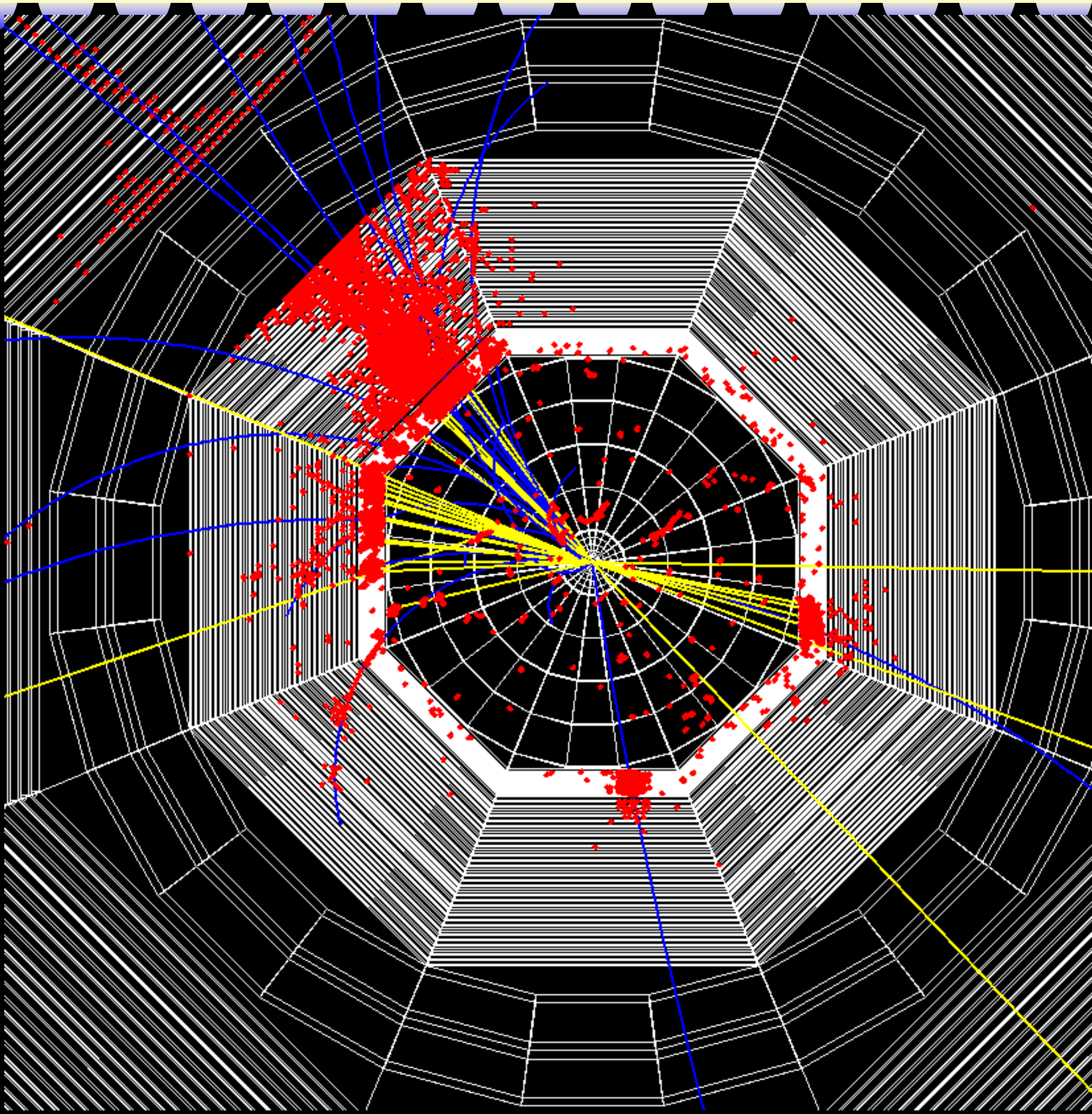


Geant4 Detector Response Simulation

- Defining detectors at runtime using a single, common executable should enable many detector variants to be simulated and compared.
- Historically, we have limited the allowed subdetector geometries to a few simplified shapes and assumed topologies for flexibility. (detparms)
- Can now do this for arbitrary detector elements using lcdd, built on top of GDML.
- Would like to bind simulation with reconstruction!

Updated Geometries

- Arbitrarily complex detectors.
- User-defined layering of stave modules.
- User-defined readout.
- Complicates reconstruction!

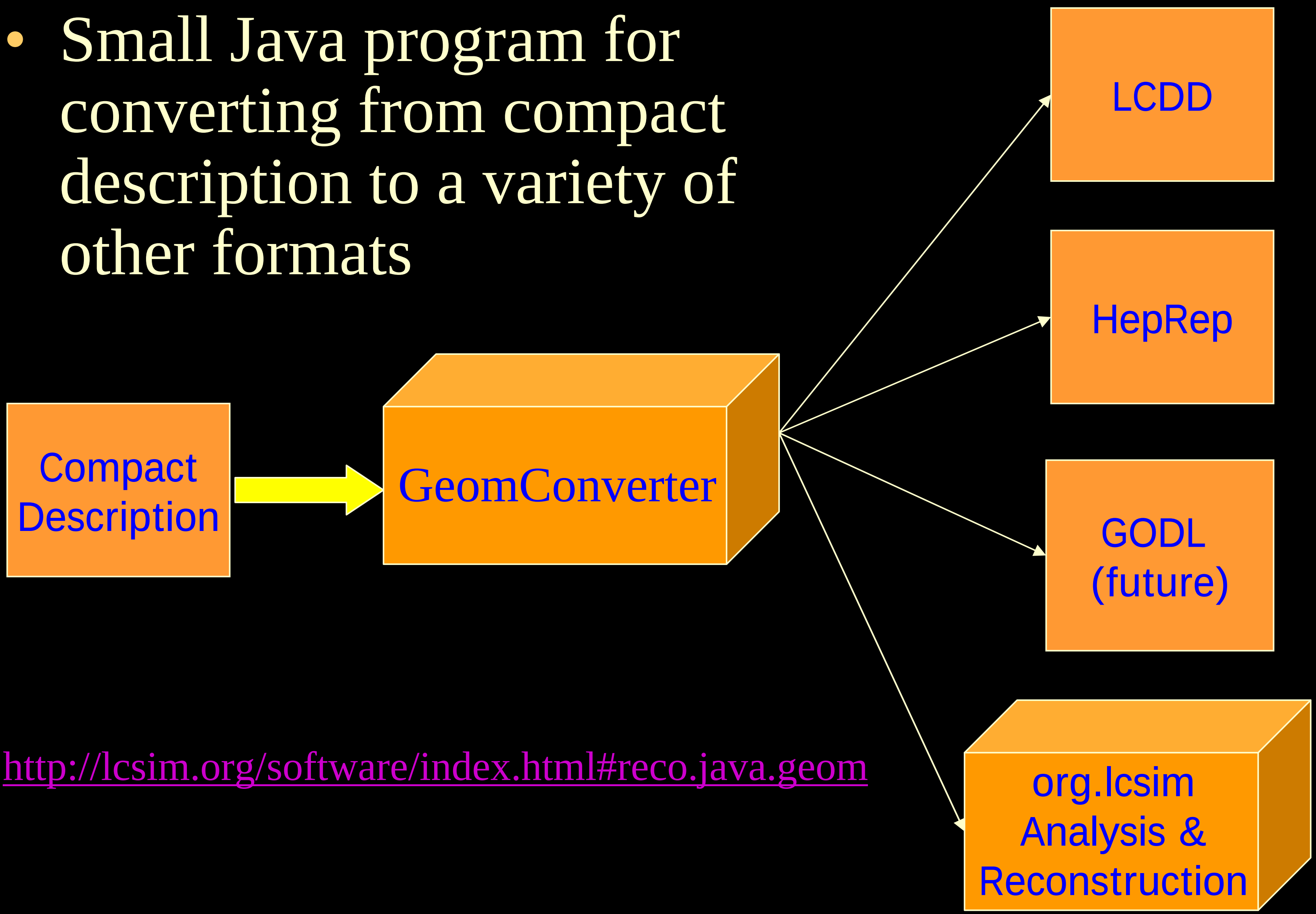


Why another geometry format?

- LCDD is great, handles any geometry, but
  - Files are large, since entry for every G4 volume
  - Simple change (e.g. # layers) may require many changes to LCDD file
  - Not right level of detail for reconstruction
- Compact format is less generic, but
  - Files are much shorter and easier to edit
  - Can handle any likely geometry/segmentation
    - May require additional “drivers” to be implemented in Java
  - Maintains XML advantages (well-formedness, editability, etc.)
  - LCDD can be generated from compact format
- Goal:
  - Rapid prototyping of detector geometries
  - Ability to provide description of new (or existing) detectors for reconstruction (org.lcsim)

GeomConverter

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



http://lcsim.org/software/index.html#reco.java.geom

Compact Elements

- <lccdd>
  - <info>
  - <define/>
  - <materials/>
  - <detectors/>
  - <readouts/>
  - <fields/>
- </lccdd>

<info>

```
<info name="sdjan03"
          author="Jeremy McCormick"
          version="1.0"
          timestamp="2004-12-13T12:00:53"
          url=“http://www.lcsim.org/detector/sdjan03">
  <comment>
   Test of the compact format for sdjan03 detector.
  </comment>
 </info>
```

<define>

```
<define>
  <constant name="cm" value="10"/>
  <!-- world -->
  <constant name="world_side" value="15000" />
  <constant name="world_x" value="world_side" />
  <constant name="world_y" value="world_side" />
  <constant name="world_z" value="world_side" />

  <!-- tracking region -->
  <constant name="tracking_region_radius" value="127.0*cm"/>
  <constant name="tracking_region_zmax" value="168.0*cm"/>

  <constant name="vertex_inner_r" value="1.2*cm"/>
  <constant name="vertex_delta_r" value="1.2*cm"/>
  <constant name="vertex_outer_z" value="12.5*cm"/>
</define>
```

- A few items are required (world_*, tracking_region_*), rest are user defined.

<materials>

```
<materials>
    <element name="Silicon_e" formula="Si" Z="14.">
        <atom value="28.09" />
    </element>

    <material name="Polystyrene">
        <D value="1.032" unit="g/cm3" />
        <composite n="19" ref="Carbon_e"/>
        <composite n="21" ref="Hydrogen_e" />
    </material>
</materials>
```

- We have a “standard” material file, need only define “special” materials.

<detectors>

```
<detectors>
  <detector id="2" name="EMBarrel" type="CylindricalCalorimeter"
                                   readout="EcalBarrHits">
     <dimensions inner_r = "127.0*cm" outer_z = "184.0*cm" />
     <layer repeat="30">
       <slice material = "Tungsten" width = "0.25*cm" />
       <slice material = "G10" width = "0.068*cm" />
       <slice material = "Silicon" width = "0.032*cm" sensitive = "yes" />
       <slice material = "Air" width = "0.025*cm" />
     </layer>
  </detector>
</detectors>
```

- Contents of detector element depends on “type”, types are extensible, see discussion later.

<readouts>

```
<readouts>
   <readout name="EcalBarrHits">
          <segmentation type="ProjectiveCylinder" thetaBins=“1000“
                                                                        phiBins=“2000"/>
          <id>layer:7,system:3,barrel:3,theta:32:11,phi:11</id>
      </readout>
</readouts>
```

- Contents of segmentation element depends on “type”, types are extensible, see discussion later.
- Support projective Barrel and Endcaps, finishing cartesian planar and fixed-z, phi cylindrical.
- IDDecoder in org.lcsim reconstruction uses same information to convert global ⇔ local

<fields>

```
<fields>
  <field type="Solenoid" name="GlobalSolenoid"
          inner_field="5.0"
          outer_field="-0.6"
          zmax="1000"
          outer_radius="144*cm+(2+1)*34*cm"/>
 </fields>
```

- Contents of field element depends on “type”, types are extensible, see discussion later.

GeomConverter Implementation

- GeomConverter provides basic functionality for reading file.
  - Plugin modules (Java classes) provide capability of generating different types of output.
  - Plugin drivers (Java classes) provide capability of supporting different types of fields, segmentations, detector shapes.
    - GeomConverter comes with a small set of generic classes for common cases (cylinders, polygonal, etc).
    - Specialized classes can be developed if necessary for strangely shaped detectors.

Dead Material

- Currently dead material can be specified as detector with no sensitive volumes.
- In future will allow dead-material to be specified using full GDML markup, included into LCDD file during generation.
  - Suitable for defining complex shapes such as masking which is normally only relevant for simulation but not reconstruction.

Getting GeomConverter

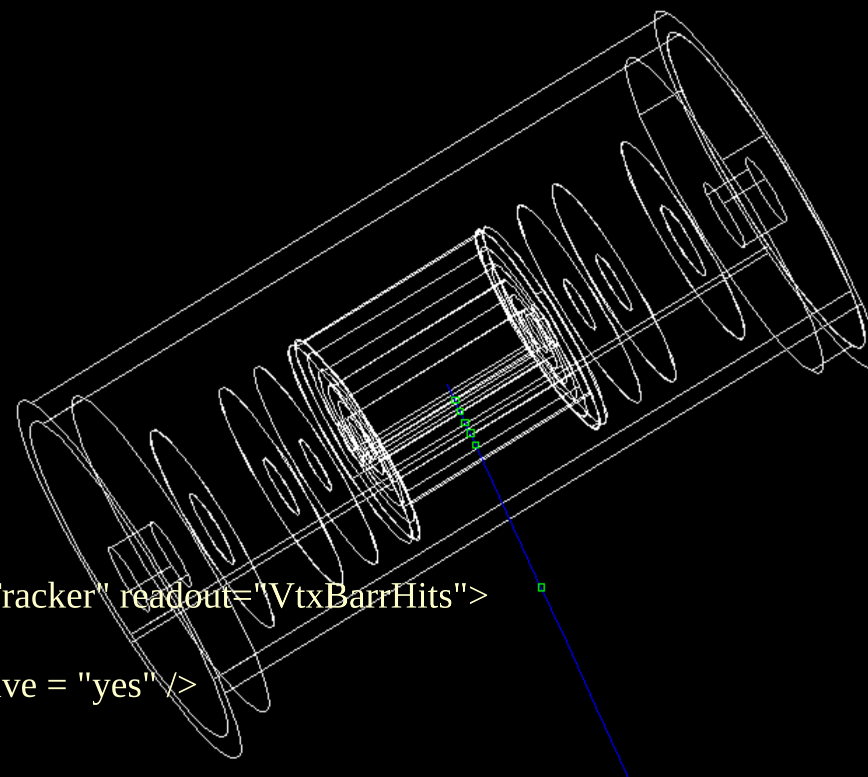
- Web Page:
  - http://www.lcsim.org/software/geomconverter
- CVS:
  - :pserver:anonymous@cvs.freehep.org:/cvs/lcd
  - module GeomConverter
  - After checkout use “maven” to build.
- GeomConverter integrates with org.lcsim reconstruction framework, see tomorrow’s talk.

Detector Variants

- Runtime XML format allows variations in detector geometries to be easily set up and studied:
  - Stainless Steel vs. Tungsten HCal sampling material
  - RPC vs. GEM vs. Scintillator readout
  - Layering (radii, number, composition)
  - Readout segmentation (size, projective vs. nonprojective)
  - Tracking detector technologies & topologies
    - TPC, Silicon microstrip, SIT, SET
    - “Wedding Cake” Nested Tracker vs. Barrel + Cap
  - Field strength
  - Far forward MDI variants (0, 2, 20 mr)

Vertex Detector

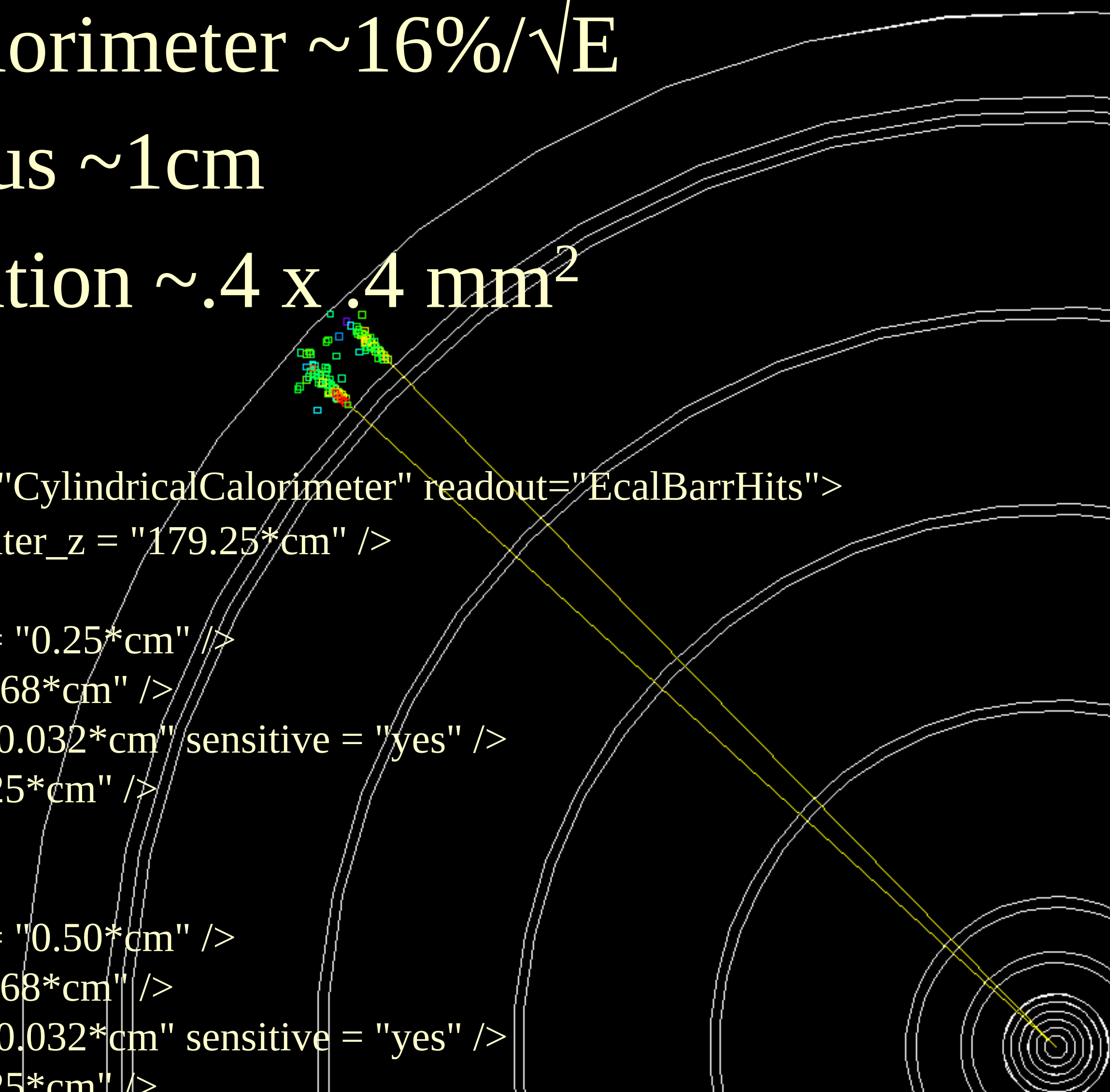
- 5 Layer CCD Barrel
- 4 Layer CCD Disks
- Be supports
- Foam Cryostat



```
<detectors>
    <detector id="0" name="BarrelVertex" type="MultiLayerTracker" readout="VtxBarrHits">
        <layer id="1" inner_r = "1.5*cm" outer_z = "6.25*cm">
            <slice material = "Silicon" width = "0.01*cm" sensitive = "yes" />
        </layer>
        <layer id="2" inner_r = "2.6*cm" outer_z = "6.25*cm">
            <slice material = "Silicon" width = "0.01*cm" sensitive = "yes" />
        </layer>
        <layer id="3" inner_r = "3.7*cm" outer_z = "6.25*cm">
            <slice material = "Silicon" width = "0.01*cm" sensitive = "yes" />
        </layer>
        <layer id="4" inner_r = "4.8*cm" outer_z = "6.25*cm" >
            <slice material = "Silicon" width = "0.01*cm" sensitive = "yes" />
        </layer>
```

EM Calorimeter

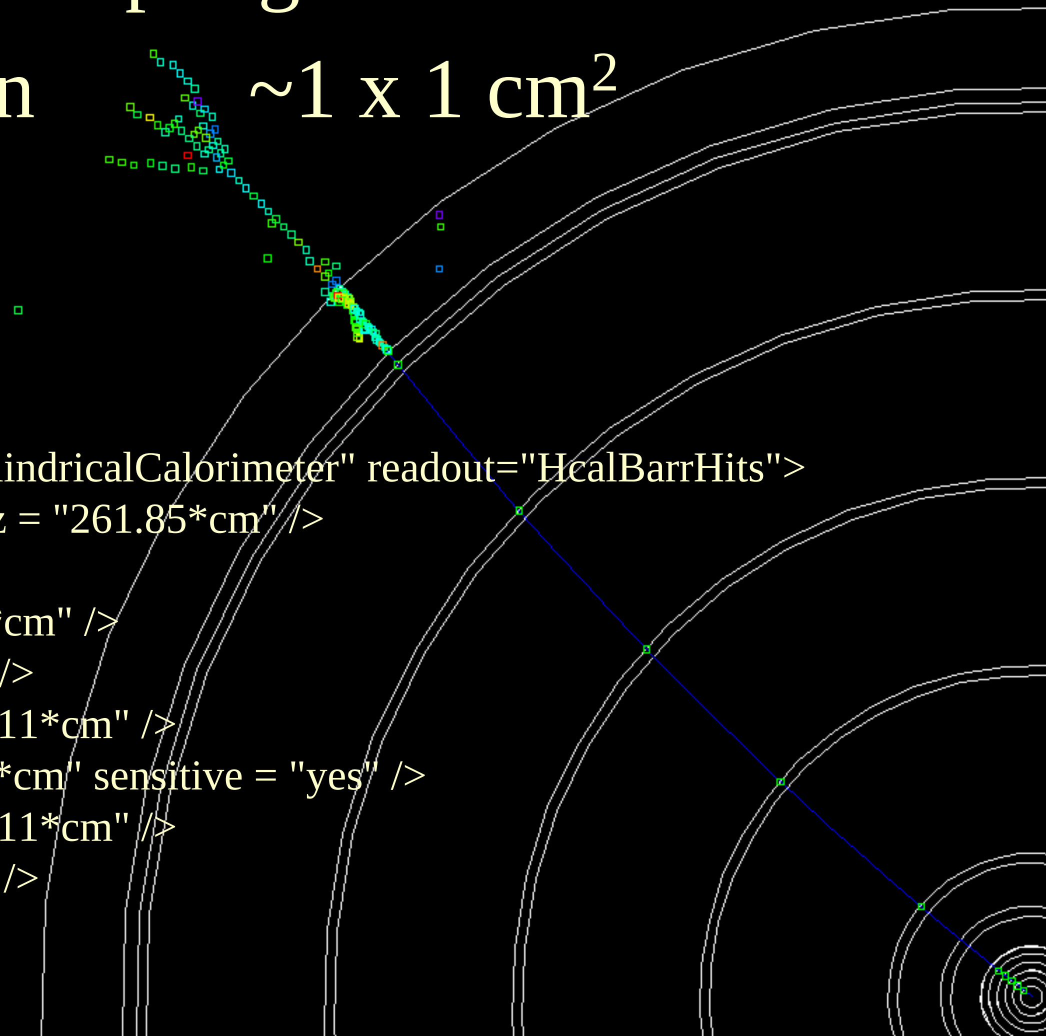
- Si-W Sampling Calorimeter ~16%/√E
- Small Moliere radius ~1cm
- Excellent segmentation ~.4 x .4 mm^2



```xml
<detector id="3" name="EMBarrel" type="CylindricalCalorimeter" readout="EcalBarrHits">
        <dimensions inner_r = "127.0*cm" outer_z = "179.25*cm" />
        <layer repeat="20">
          <slice material = "Tungsten" width = "0.25*cm" />
          <slice material = "G10" width = "0.068*cm" />
          <slice material = "Silicon" width = "0.032*cm" sensitive = "yes" />
          <slice material = "Air" width = "0.025*cm" />
        </layer>
<layer repeat="20">
          <slice material = "Tungsten" width = "0.50*cm" />
          <slice material = "G10" width = "0.068*cm" />
          <slice material = "Silicon" width = "0.032*cm" sensitive = "yes" />
          <slice material = "Air" width = "0.025*cm" />
```

Hadronic Calorimeter

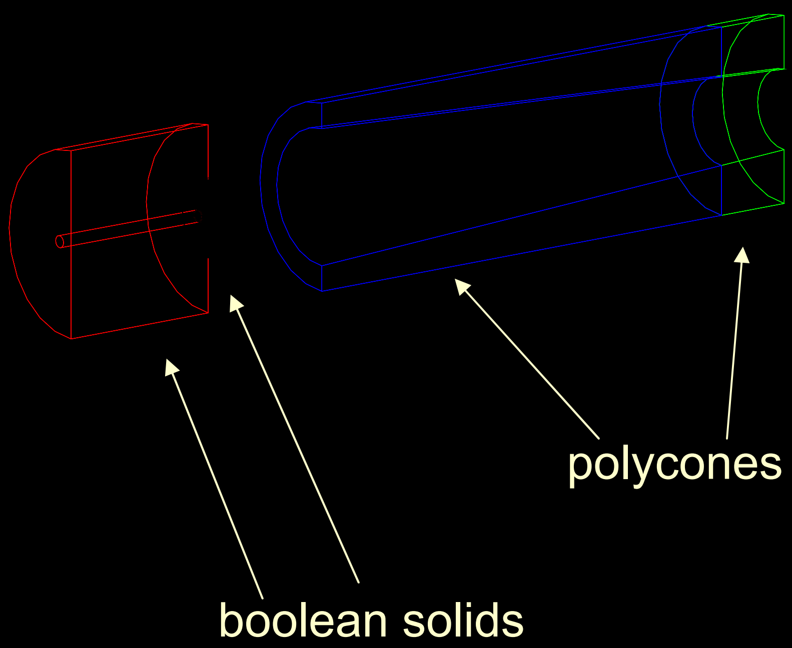
- W(SS)+RPC (Scint.) Sampling
- Excellent segmentation ~1 x 1 cm^2

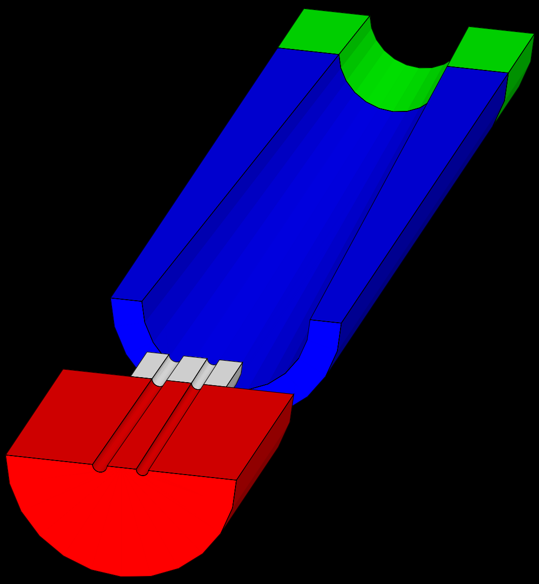


```
<detector id="3" name="HADBarrel" type="CylindricalCalorimeter" readout="HcalBarrHits">
      <dimensions inner_r = "138.26*cm" outer_z = "261.85*cm" />
      <layer repeat="55">
        <slice material = "Tungsten" width = "0.7*cm" />
        <slice material = "G10" width = "0.3*cm" />
        <slice material = "PyrexGlass" width = "0.11*cm" />
        <slice material = "RPCGas" width = "0.12*cm" sensitive = "yes" />
        <slice material = "PyrexGlass" width = "0.11*cm" />
        <slice material = "Air" width = "0.16*cm" />
      </layer>
  </detector>
```

MDI - BDS

Have defined compact detector elements which allow forward masks and detector elements to be described. Will provide 2mr and 20mr solutions for studies.





Detector Repository

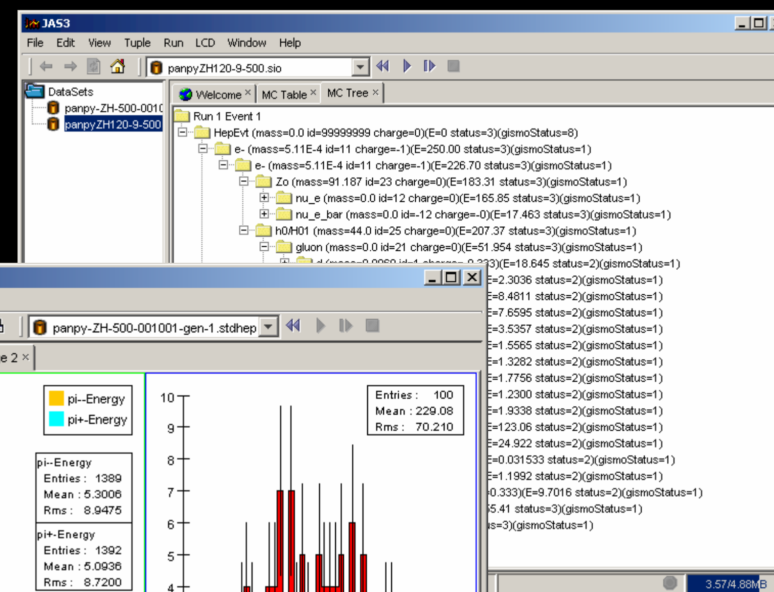
- Standard detector descriptions are available in the LCDetectors package
- CVS:
  - :pserver:anonymous@cvs.freehep.org:/cvs/lcd
  - module LCDetectors
- Currently have sdjan03, sdfeb05, sidmay05*
- Plan to support GLD and LDC.
  - Need help in defining geometries!
- Also have a template for new designs.

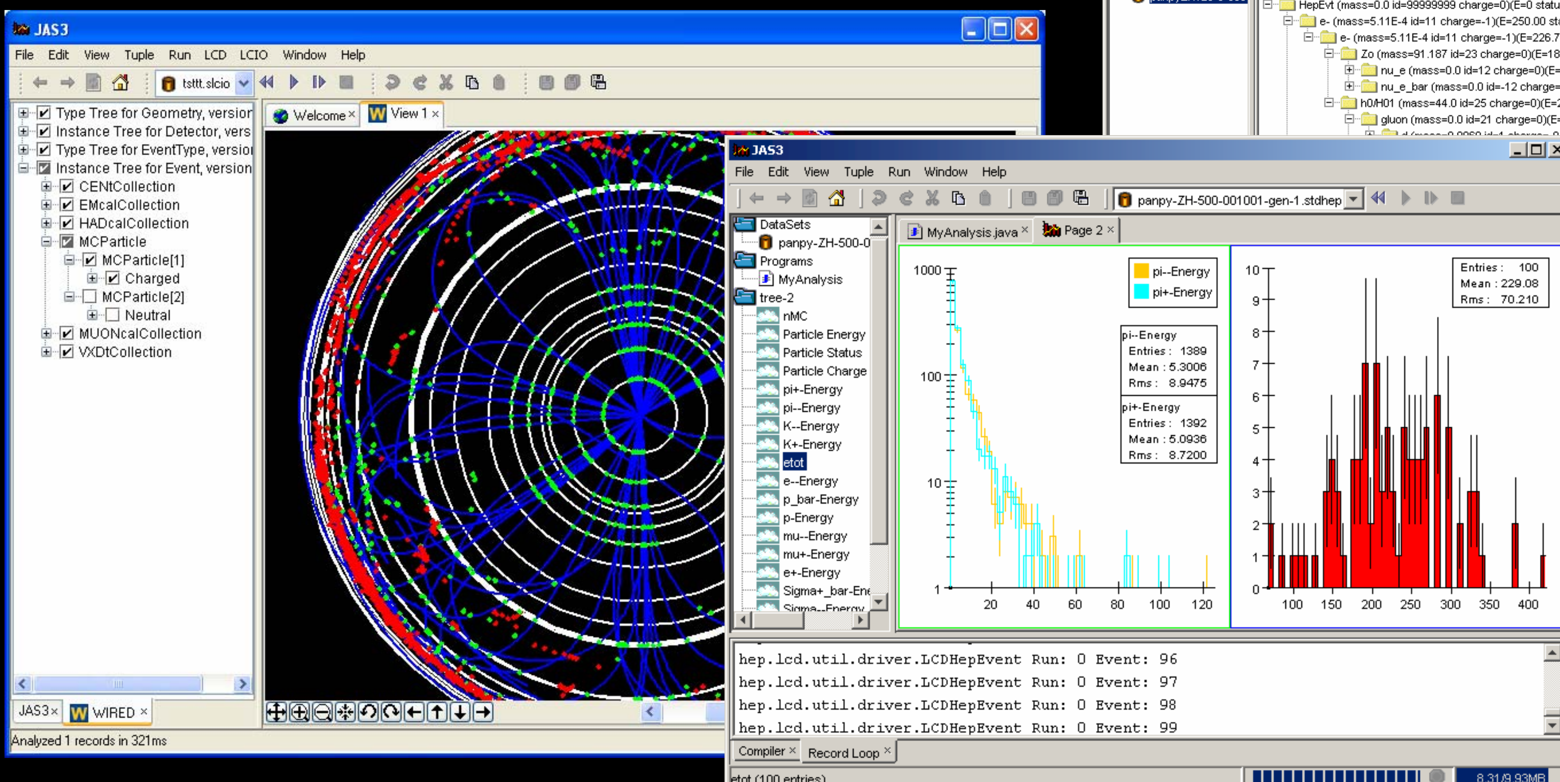
Detector Description Summary

- Compact detector description provides not only a simpler definition of the detector, but also a binding for the visualization and the reconstruction.
- First release of GeomConverter now available
  - Tested with sdjan03, sdfeb05, sidmay05*
    - Able to generate full LCDD description for SLIC
    - Able to generate HepRep for display with WIRED
  - Encourage others to define variants or other concepts.
- Will continue to enhance in parallel with org.lcsim reconstruction package.

Reconstruction/Analysis

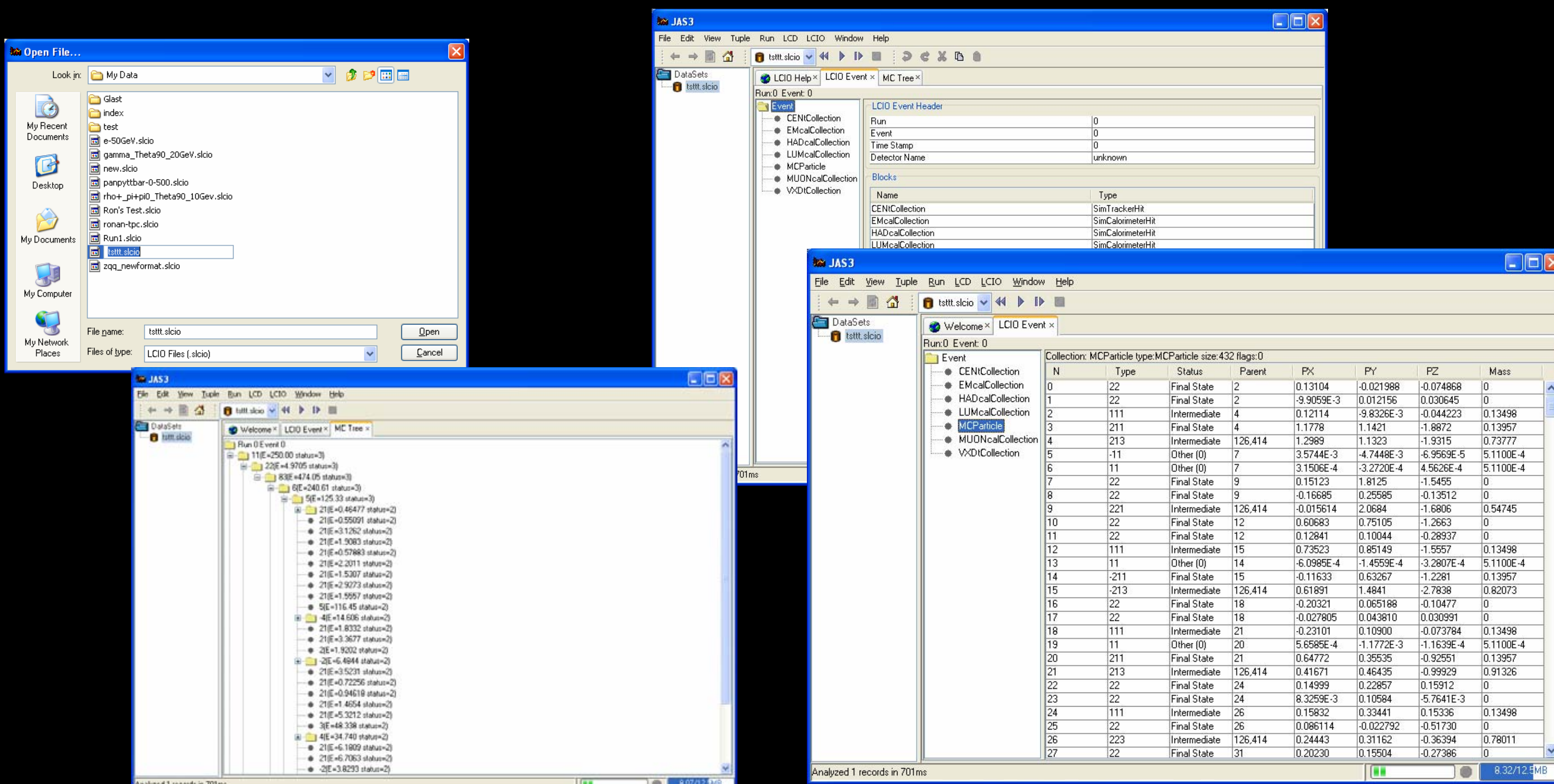
- Java Analysis Studio (JAS) provides a framework for event visualization (with WIRED) , reconstruction & analysis.





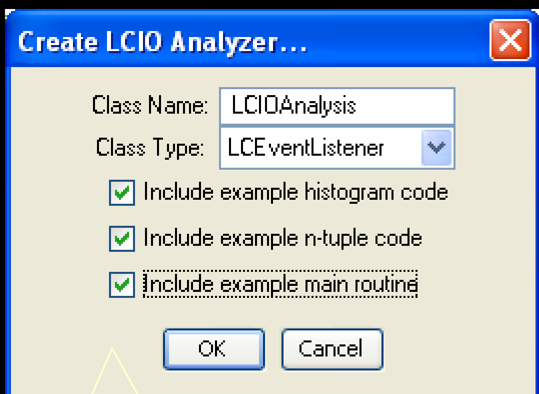
LCIO plugin for JAS3

- Works with any LCIO file
- Diagnostic tools allow to step through and view events

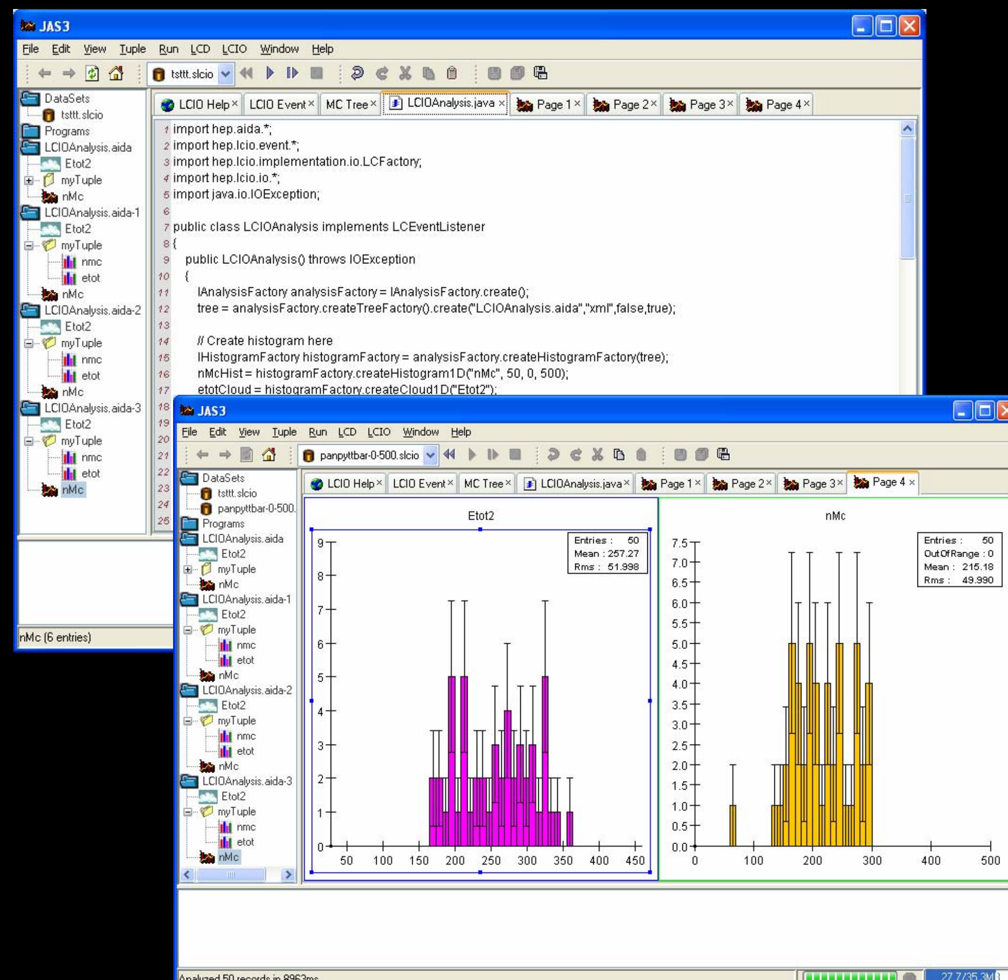


LCIO plugin for JAS3

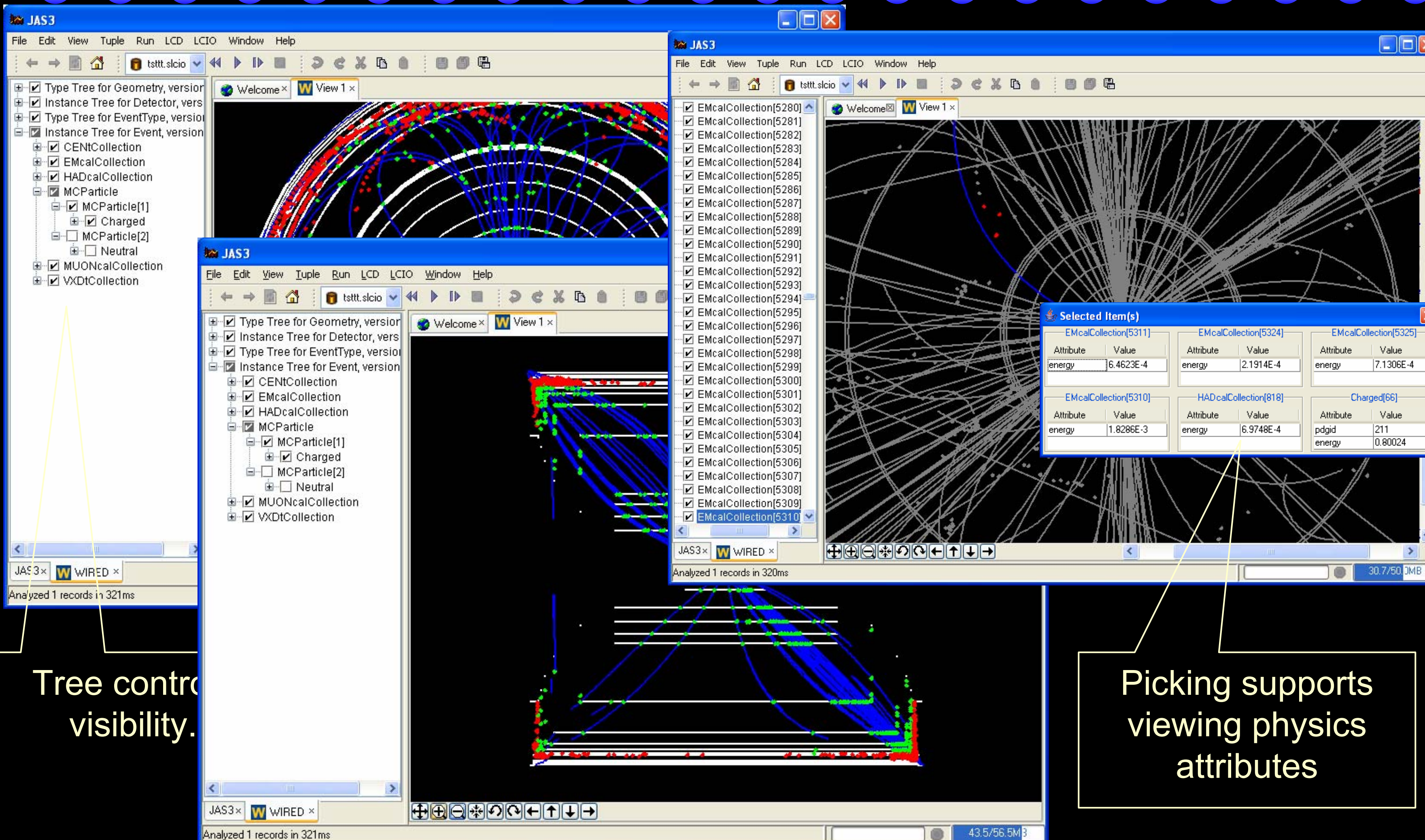
- Event Analysis



Wizard will create outline of analysis, and can include sample analysis code. Main routine allows running outside of JAS



WIRED with LCIO Plugin



Tree contr
visibility.

Picking supports viewing physics attributes

org.lcsim plugin for JAS3

- Provides access to the full software suite.
- Autoupdates with new releases (~weekly).
- Comes with a set of example programs to speed introduction to the reconstruction and analysis:

SimpleGenerator.java	Simple diagnostic event generator.
Simple.java	Simply prints the event header of each event analyzed.
Analysis101.java	Intro to analysis with AIDA.
SimpleFastMC.java	Running the Fast MC.
SimpleOutput.java	Example of writing LCIO output
JetFinding.java	Using the Jet Finder
ClusterFinding.java	Cluster Finding example

Reconstruction/Analysis Overview

- Java based reconstruction and analysis package
  - Runs standalone or inside Java Analysis Studio (JAS)
  - Fast MC → Smeared tracks and calorimetry clusters
  - Full Event Reconstruction
    - detector readout digitization (CCD pixels & Si μ-strips)
    - *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 (Vertex Finding, Jet Finding, Flavor Tagging)
  - Beam Background Overlays at detector hit level
- Very aggressive program, strong desire to "do it right."

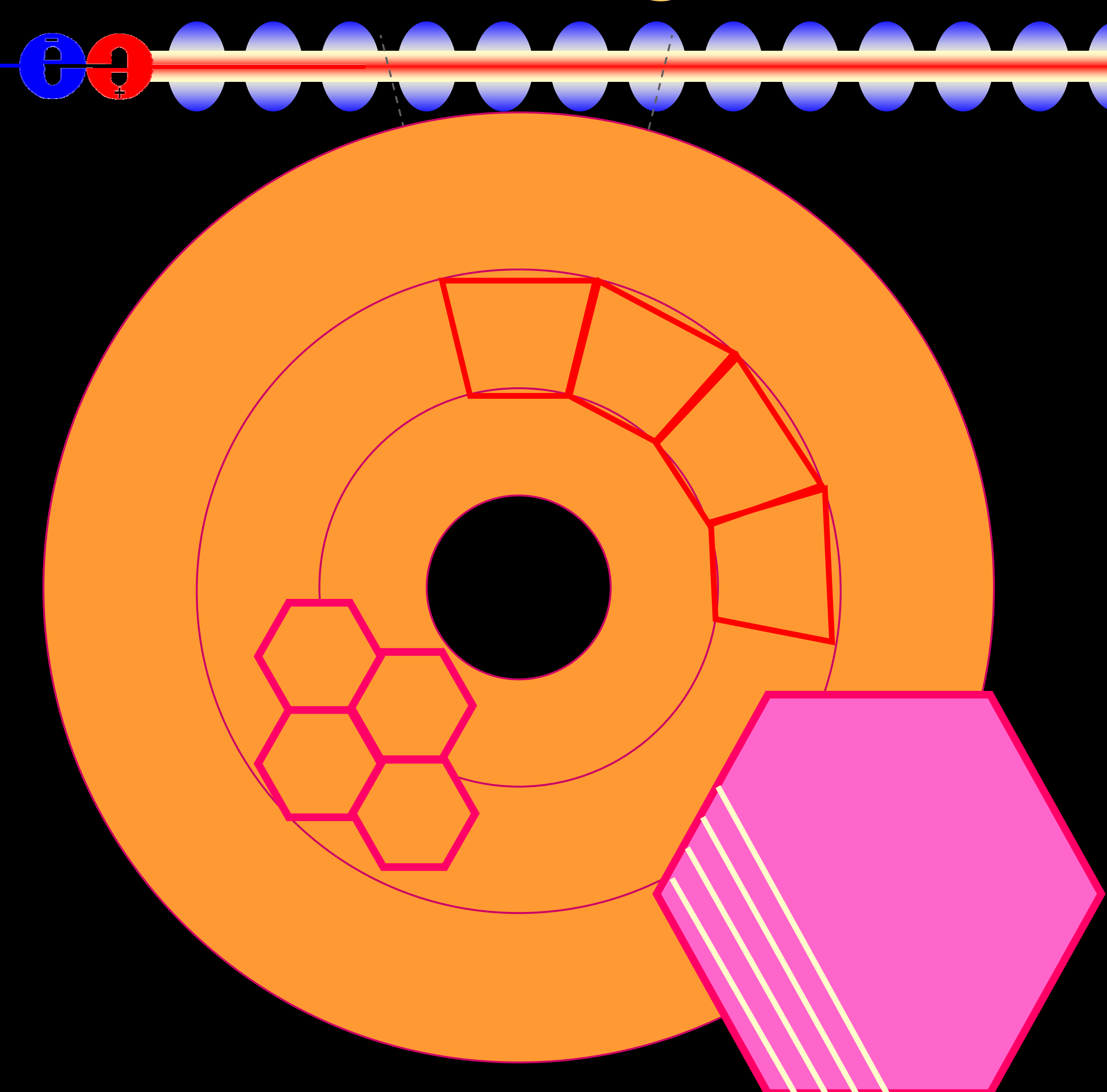
Tracking Detector Readout

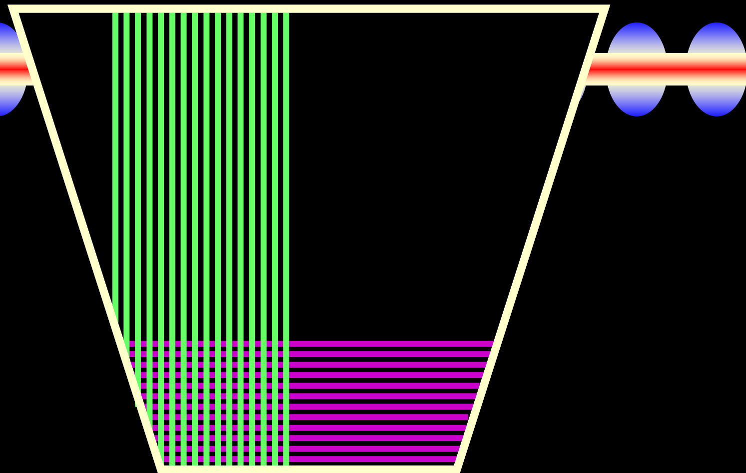
- Hits in Trackers record full MC information.
- Digitization is deferred to analysis stage.
- Nick Sinev has released a package to convert hits in silicon to CCD pixel hits.

  MC Hits$\rightarrow$ Pixels & PH$\rightarrow$ Clusters$\rightarrow$ Hits ($x \pm \delta x$)
- UCSC developed long-shaping-time μ-strip sim.

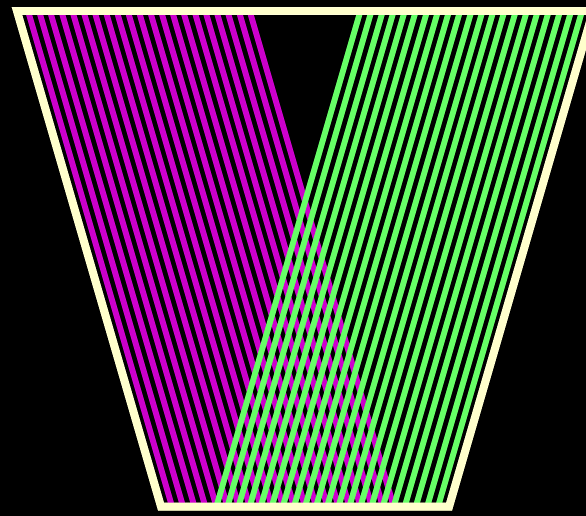
  MC Hits$\rightarrow$ Strips & PH$\rightarrow$ Clusters$\rightarrow$ Hits ($\varphi \pm \delta\varphi$)
- SLAC developing short-strip simulation.
- Correctly study occupancies, overlaps, ghost hits.

Tiling Forward Disks





Large Angle Stereo

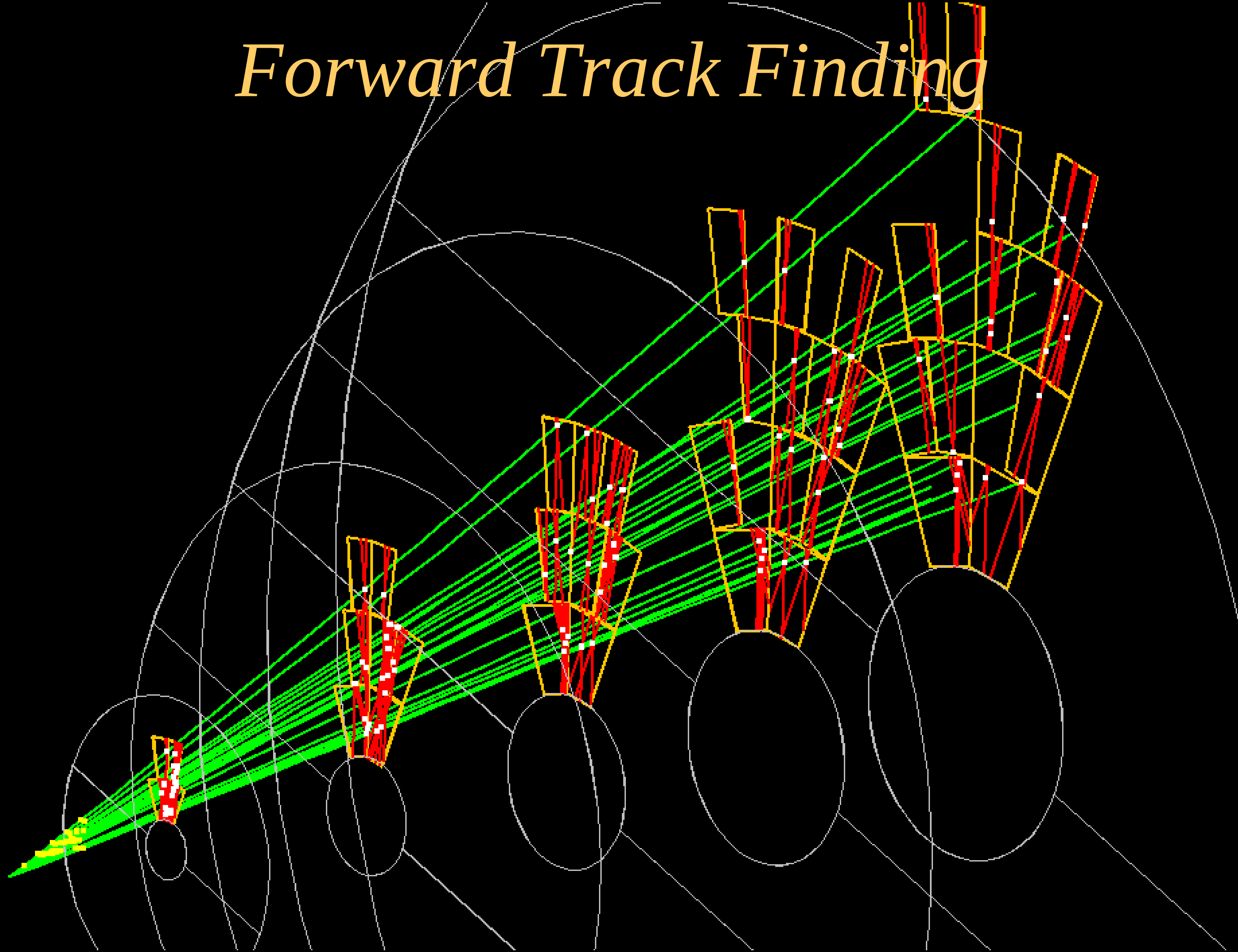


Shallow Angle Stereo

Track Finding

- Nick Sinev has released standalone pattern recognition code for the 2D Barrel VXD hits.
  - High efficiency, even in presence of backgrounds.
  - Efficient at low momentum.
  - Propagates tracks into Central Tracker to pick up φ hits
- Conformal-mapping pattern recognition also available. Fast, but not yet tuned (97% vs 99+%).
- Work also ongoing to find MIP stubs in Cal and propagate inwards to tracker (Kansas State).

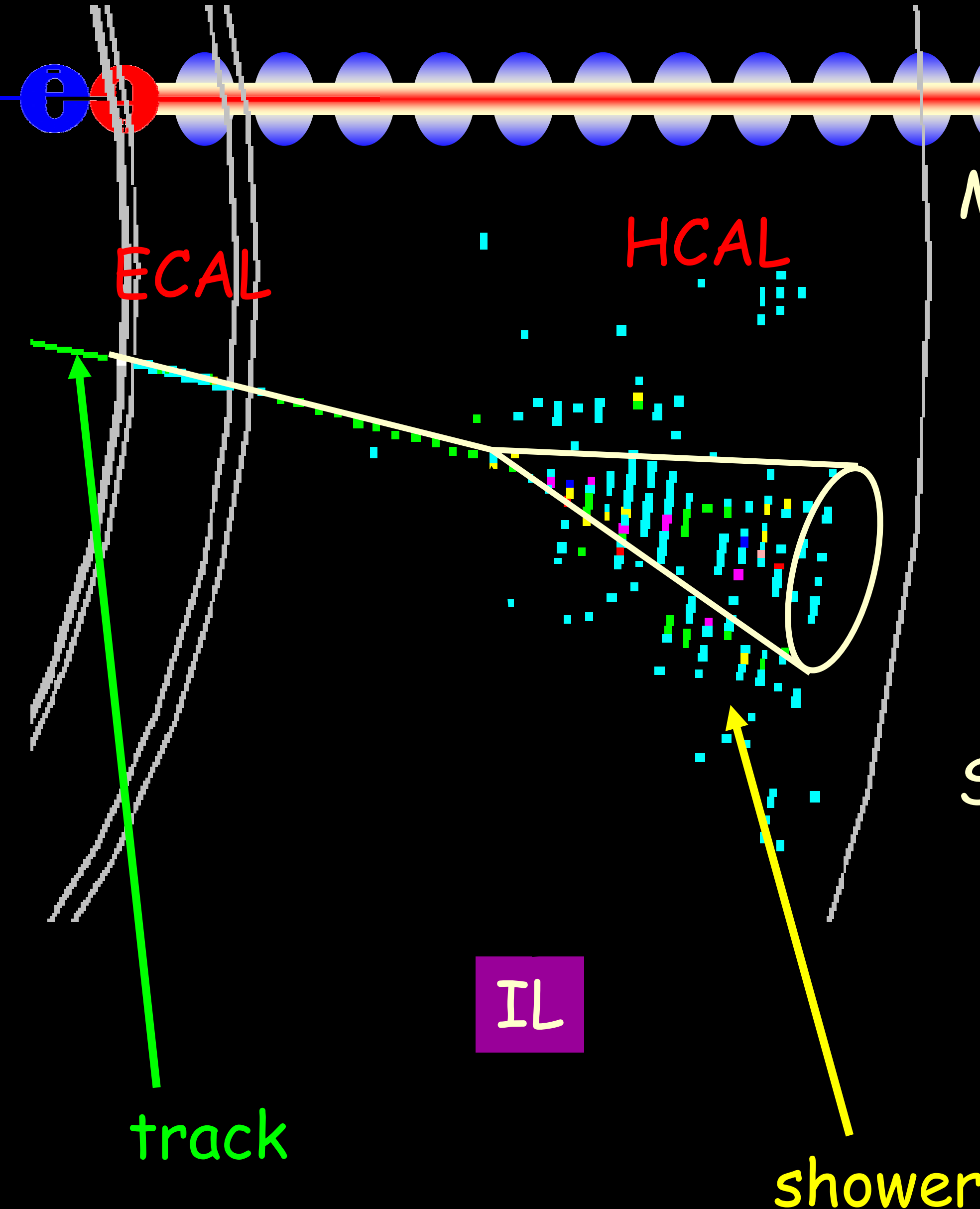
Forward Track Finding



Calorimeter Reconstruction

- A number of groups are following different approaches towards individual particle reconstruction ("particle flow")
  - SLAC, Argonne, NICADD, Kansas State, Iowa, FNAL…
- Identifying photon, electron, charged & neutral hadron showers and muons in the calorimeter.
- Tracking in the calorimeter can assist pattern recognition in the trackers!

Shower reconstruction by track extrapolation



Mip reconstruction :

Extrapolate track through CAL layer-by-layer

Search for "Interaction Layer"

-> Clean region for photons (ECAL)

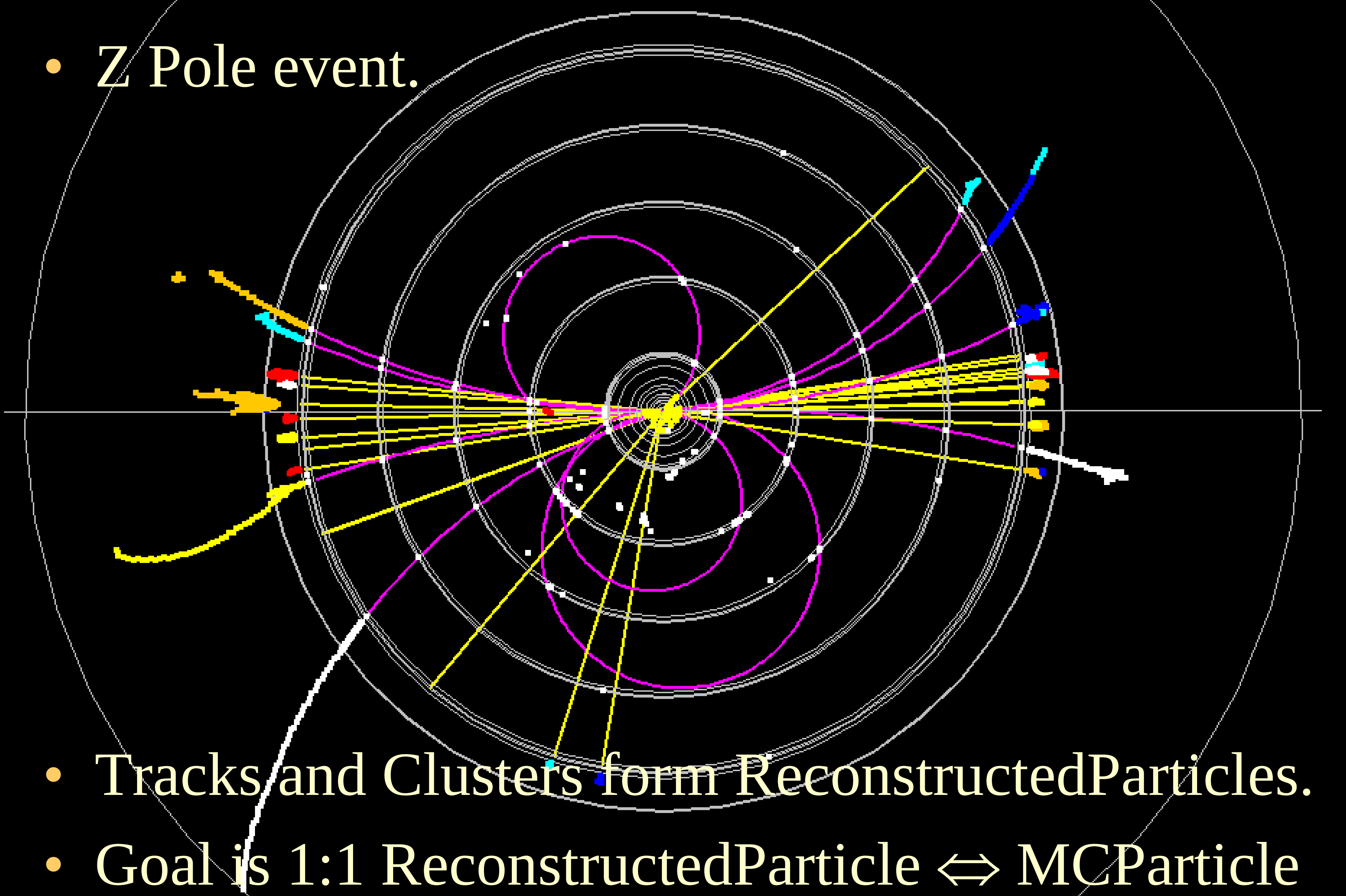
Shower reconstruction :

Define tubes for shower in ECAL, HCAL after IL

Optimize, iterating tubes in E,HCAL separately (E/p test)

Individual Particle Reconstruction

- Z Pole event.



- Tracks and Clusters form ReconstructedParticles.
- Goal is 1:1 ReconstructedParticle ⇔ MCParticle

Simulation Summary

- ALCPG sim/reco supports an ambitious international simulation effort with a very small group of people.
- Provides full data samples for ILC physics studies.
- Provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description.
- Reconstruction & analysis framework maturing, additional manpower maps directly to physics and detector results.
- Aim to characterize/optimize performance of the Detector Concepts for CDR starting at Snowmass.

Additional Information

- Linear Collider Simulations
  - http://www.lcsim.org
- ALCPG Simulation/Reconstruction
  - http://confluence.slac.stanford.edu/display/ilc/Home
- Discussion Forums
  - http://forum.linearcollider.org