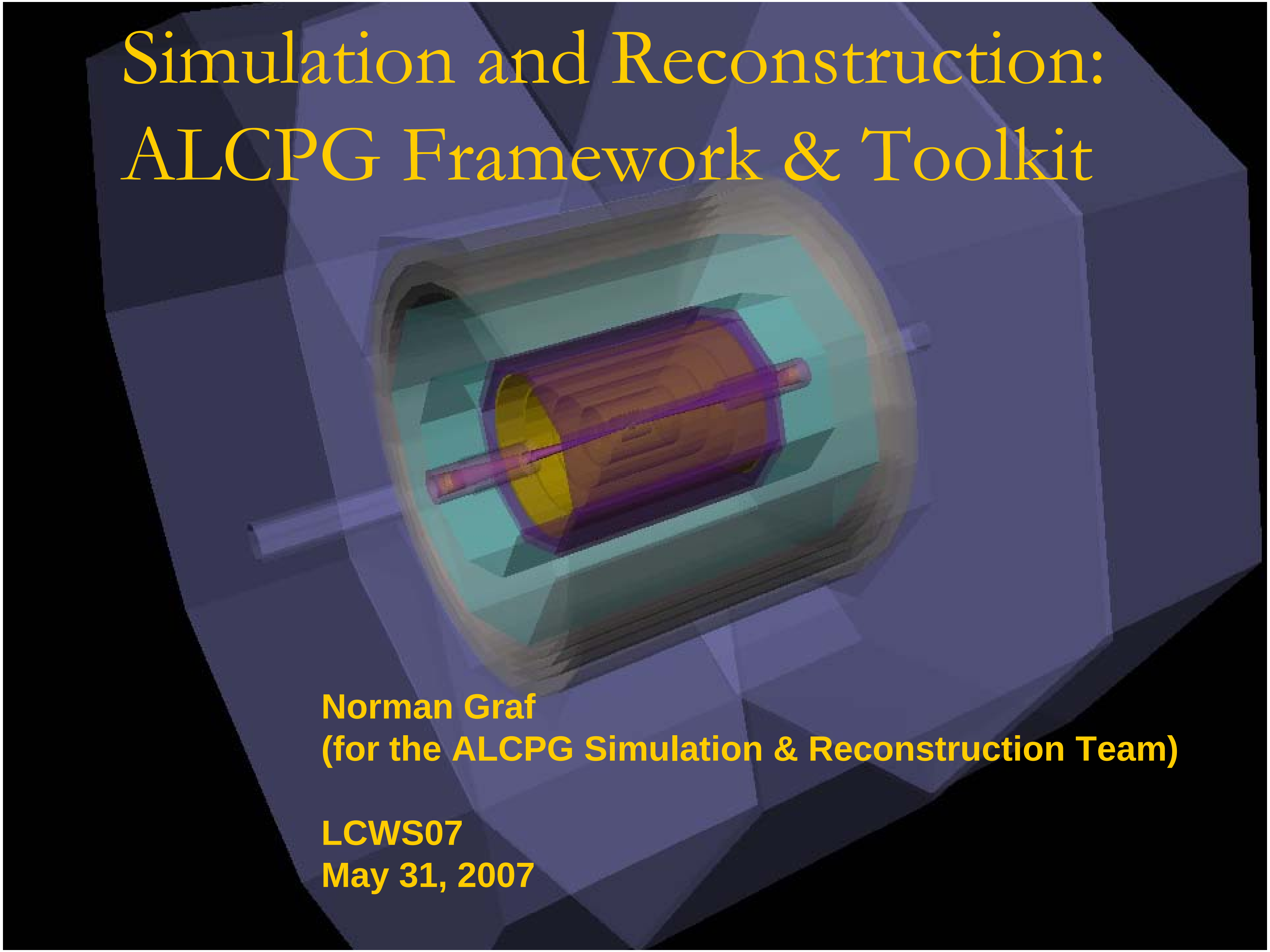
Simulation and Reconstruction: ALCPG Framework & Toolkit



Norman Graf
(for the ALCPG Simulation & Reconstruction Team)

LCWS07
May 31, 2007

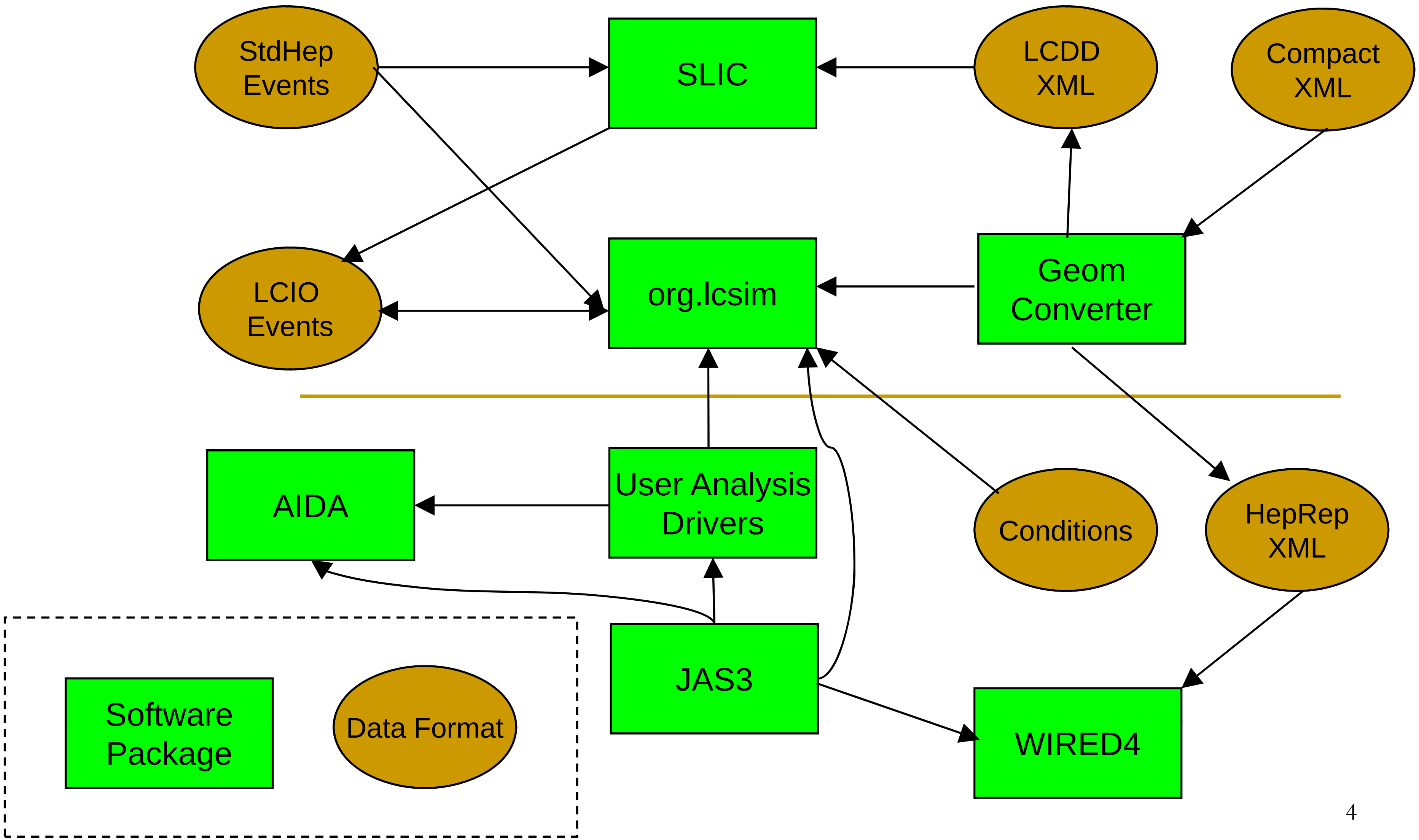
Introduction

- This talk not meant to be an in-depth summary of all existing functionality.
  - Not enough time.
  - Been done many times in the past.
- Simply an update on some recent, added functionality.
- Geometry system improvements.
- Improvements to “easy” detector simulations (i.e. via compact.xml).
  - Si wafers, TPC simulation with cuts by region
  - lelaps for fast MC hits generation.
  - polyhedral Calorimeters
- Event samples, both signal and backgrounds.

Goals

- Enable full studies of ILC physics to optimize detector design and eventual physics output
  - Use realistic detector geometries
  - Full simulation (in combination with fast parameterized MCs)
  - Full reconstruction
    - Simulate benchmark physics processes on different full detector designs.
    - Encourage development of realistic analysis algorithms
    - See how these algorithms work with full detector simulations
- Facilitate contribution from physicists in different locations with various amounts of time available (normally not much!)
  - Software should be easy to install, learn, use
    - Goal is to allow software to be installed from CD or web with no external dependencies
    - Support via web based forums, tutorials, meetings.

Overview: "SiD/ALCPG" Framework

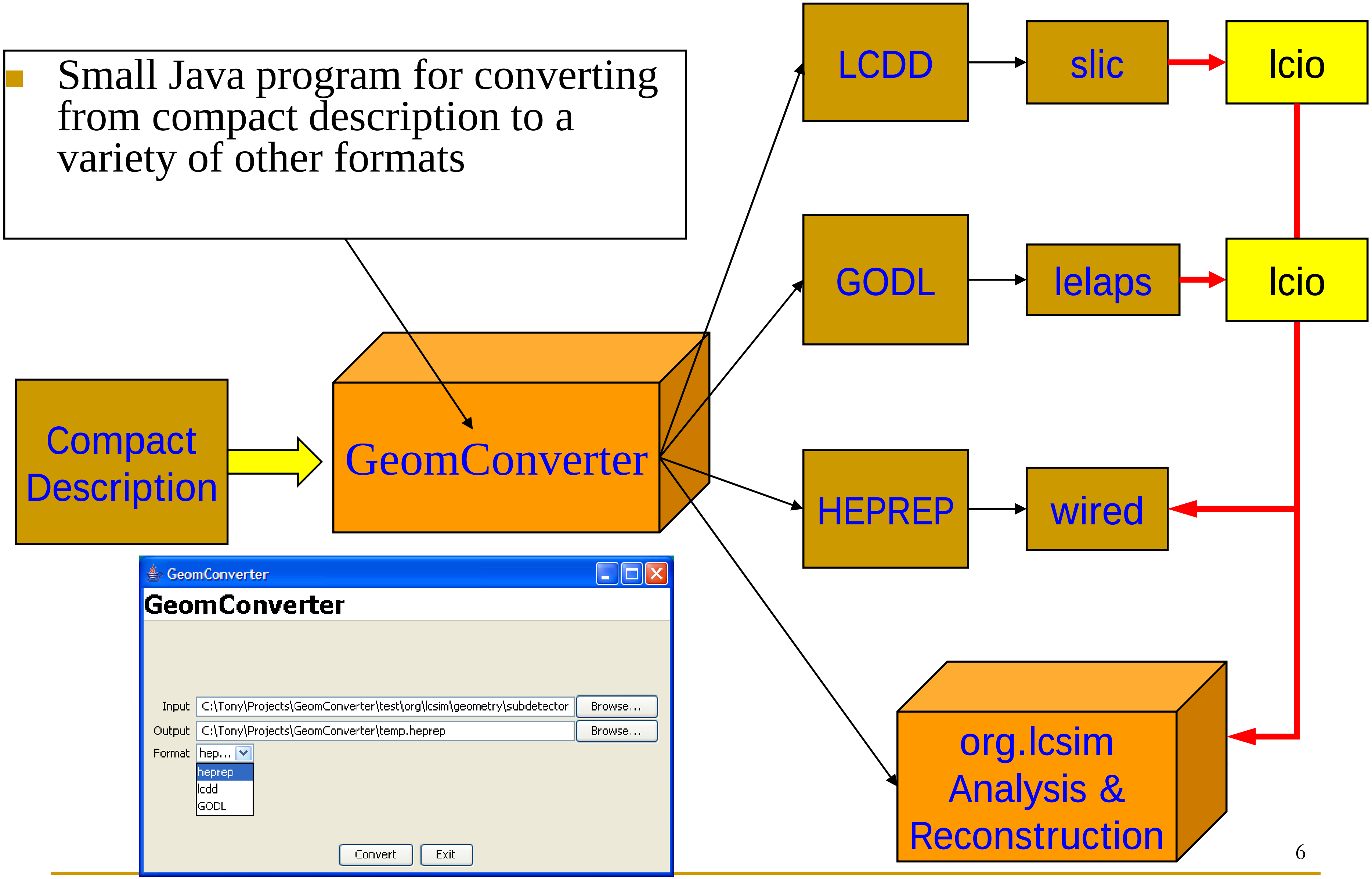


Improved Detector Simulations

- The full simulation package slic reads in geometries in lcdd, which is a low-level format that targets Geant4 primitives.
  - Detectors of arbitrarily complex shape and readout can be simulated using only xml file as input.
- However, it would be extremely tedious to generate these files.
- Would also not provide a connection to the reconstruction, nor to the event display.
- Prefer (but not required) to define geometries using a “compact” description.
- Small Java program for converting from compact description to a variety of other formats.
  - GeomConverter.

GeomConverter

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



Geometry updates in org.lcsim

- hierarchical detector model & geometry model
- parameters & identifiers
- solids & materials
- navigation, point location
- logical and physical volumes
- readout
- coordinate transformation
  - local to global
  - global to local
  - parent to local

Detailed Geometry in org.lcsim

- Geometry tree
  - **hierarchy of PhysicalVolumes and LogicalVolumes**
    - **LogicalVolume**
      - **shapes – parameters, isInside**
      - **materials - A, Z, density, radiation length, interaction length, etc.**
    - **PhysicalVolume**
      - **transformation - translation + rotation**
- DetectorElement tree **–**
  - **hierarchy of DetectorElements with uplinks**
    - **What DetectorElement is point inside?**
    - **What position of a DetectorElement?**
    - **What is the global to local coordinate transformation for the DetectorElement?**
- **Existing** Detector, Subdetector **become DetectorElements**

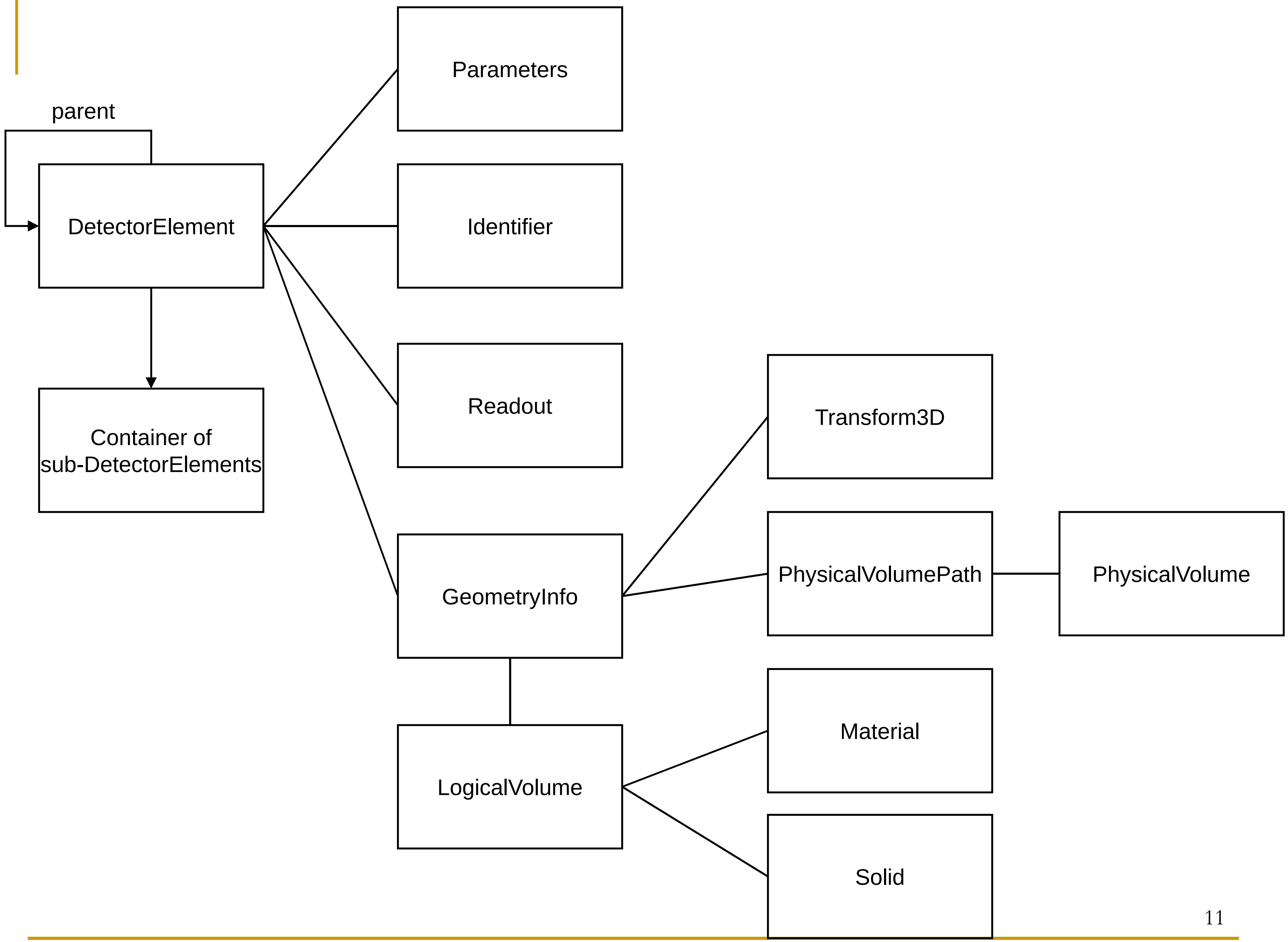
```java
// Get child DetectorElements of the Detector.
IDetectorElementContainer detElems = detector.getChildren();
// Loop over the child DEs.
for ( IDetectorElement de : detElems )
{
   // Print the name.
   System.out.println( de.getName() );
   // Print the position.
   if ( de.hasGeometryInfo() )
   {
      System.out.println( de.getGeometry().getPosition() );
   }
   // Print the names of the children.
   for ( IDetectorElement child : de.getChildren() )
   {
      System.out.println( "  " + child.getName() );
   }
}
```

Use cases

- Tracker
  - detailed geometry studies
    - module/sensor pitch, thickness, materials, components, alignment
  - digitization
    - exact sensor locations and detailed geometry information
  - conversion of components to cylinders and planes for trf propagators
  - dE/dx, radiation length, nuclear interaction length calculations
- Calorimeter
  - clustering in complex geometries (neighbor finding)
  - readout of planar pads in staves
  - modeling of realistic calorimeter detector geometries
  - attachment of conditions information to the appropriate parts/subparts
  - access to the detailed detector description from subdetector
- Detector development
  - writing generic detector output drivers (volumes, materials, shapes, etc.)
  - converting to/from a detailed representation for simulation, event display, geometry model (e.g. for reconstruction)

Features

- Computation of 3D rotations and translations using CoordinateTransformation3D
- Access to complete geometry tree using the navigator
- Lookup by name
  - DetectorElement, LogicalVolume, PhysicalVolume, Solid, Material, etc.
- Locate the leaf geometry node from a global point to given depth
- Global to local transformation of DetectorElements
- Parent to local transformation of nested DetectorElements
- Volume and subdetector center position
- Detailed materials information and API
- IGeometryInfo
  - center position of a DetectorElement
  - transformation to a DetectorElement from the global coordinate system
  - cached transformations (in progress)
    - global to local
    - local to global
    - parent to local


parent
DetectorElement
Container of
sub-DetectorElements
Parameters
Identifier
Readout
GeometryInfo
LogicalVolume
Transform3D
PhysicalVolumePath
PhysicalVolume
Material
Solid

Layer Example

```
IDetectorElement ecal = DetectorElementStore.getInstance().find("EcalBarrel");

for (IDetectorElement layer : ecal.getChildren())
{
        IGeometryInfo geom = layer.getGeometry();
        Transform3D globalToLocal = geom.getLocalToGlobal();
        ILogicalVolume lv = geom.getLogicalVolume();
        ISolid solid = lv.getSolid();
        Tube tube = (Tube)solid;
        double innerRadius = tube.getInnerRadius();
        IMaterial material = lv.getMaterial();
        IPhysicalVolume pv = geom.getPhysicalVolume();
        int copyNo = pv.getCopyNumber();
        Readout readout = layer.getReadout();
        List<CalorimeterHit> hits = readout.getHits(CalorimeterHit.class);
        Identifier id = layer.getIdentifier();
}
```

Silicon Tracking Detectors

- For the purposes of quickly scanning the parameter space of number of tracking layers and their radial and z positioning, etc. have been simulating the trackers as cylindrical shells or planar disks.
- Are now moving beyond this to be able to realistically simulate buildable subdetectors.
- Have always been able to simulate arbitrarily complex shapes in slic using lcdd, but this is a very verbose format.
- Introduced Geometry and Detector Element trees to handle arbitrary hierarchies of detector elements.
- Have now introduced tilings of planar detectors (simulating silicon wafers) into the compact xml description.

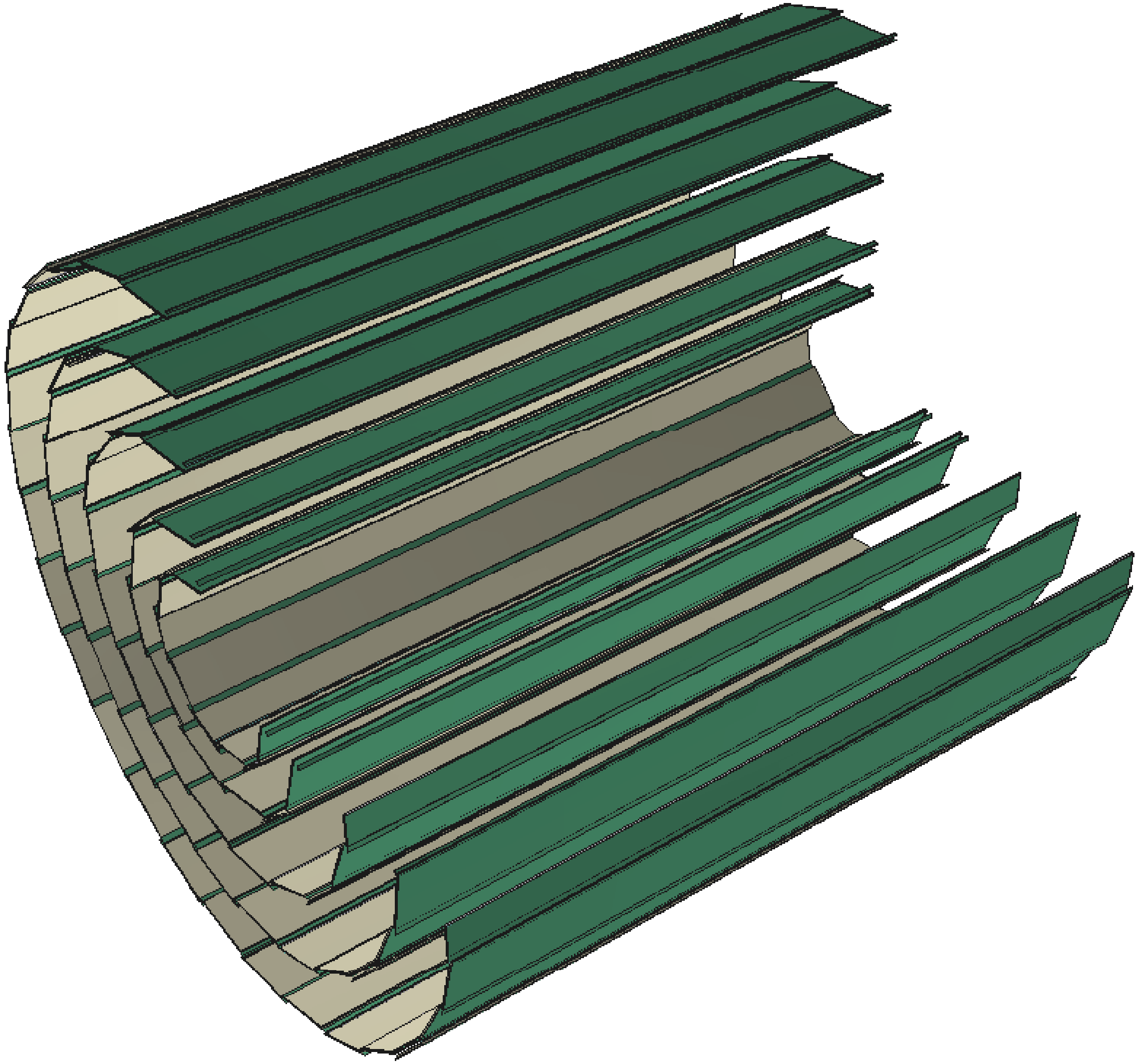
xml: Defining a Module

```
<module name="VtxBarrelModuleInner">
   <module_envelope width="9.8" length="63.0 * 2" thickness="0.6"/>
   <module_component width="7.6" length="125.0" thickness="0.26"
                                 material="CarbonFiber" sensitive="false">
                              <position z="-0.08"/>
   </module_component>
   <module_component width="7.6" length="125.0" thickness="0.05"
                                material="Epoxy" sensitive="false">
                             <position z="0.075"/>
   </module_component>
   <module_component width="9.6" length="125.0" thickness="0.1"
                                 material="Silicon" sensitive="true">
                              <position z="0.150"/>
   </module_component>
</module>
```

xml: Placing the modules

```
<layer module="VtxBarrelModuleInner" id="1">
        <barrel_envelope inner_r="13.0" outer_r="17.0" z_length="63 * 2"/>
        <rphi_layout phi_tilt="0.0" nphi="12" phi0="0.2618" rc="15.05" dr="-1.15"/>
        <z_layout dr="0.0" z0="0.0" nz="1"/>
</layer>
<layer module="VtxBarrelModuleOuter" id="2">
        <barrel_envelope inner_r="21.0" outer_r="25.0" z_length="63 * 2"/>
        <rphi_layout phi_tilt="0.0" nphi="12" phi0="0.2618" rc="23.03" dr="-1.13"/>
        <z_layout dr="0.0" z0="0.0" nz="1"/>
</layer>
<layer module="VtxBarrelModuleOuter" id="3">
        <barrel_envelope inner_r="34.0" outer_r="38.0" z_length="63 * 2"/>
        <rphi_layout phi_tilt="0.0" nphi="18" phi0="0.0" rc="35.79" dr="-0.89"/>
        <z_layout dr="0.0" z0="0.0" nz="1"/>
</layer>
<layer module="VtxBarrelModuleOuter" id="4">
        <barrel_envelope inner_r="46.6" outer_r="50.6" z_length="63 * 2"/>
        <rphi_layout phi_tilt="0.0" nphi="24" phi0="0.1309" rc="47.5" dr="0.81"/>
        <z_layout dr="0.0" z0="0.0" nz="1"/>
</layer>
<layer module="VtxBarrelModuleOuter" id="5">
        <barrel_envelope inner_r="59.0" outer_r="63.0" z_length="63 * 2"/>
        <rphi_layout phi_tilt="0.0" nphi="30" phi0="0.0" rc="59.9" dr="0.77"/>
        <z_layout dr="0.0" z0="0.0" nz="1"/>
</layer>
```

The Barrel Vertex Detector

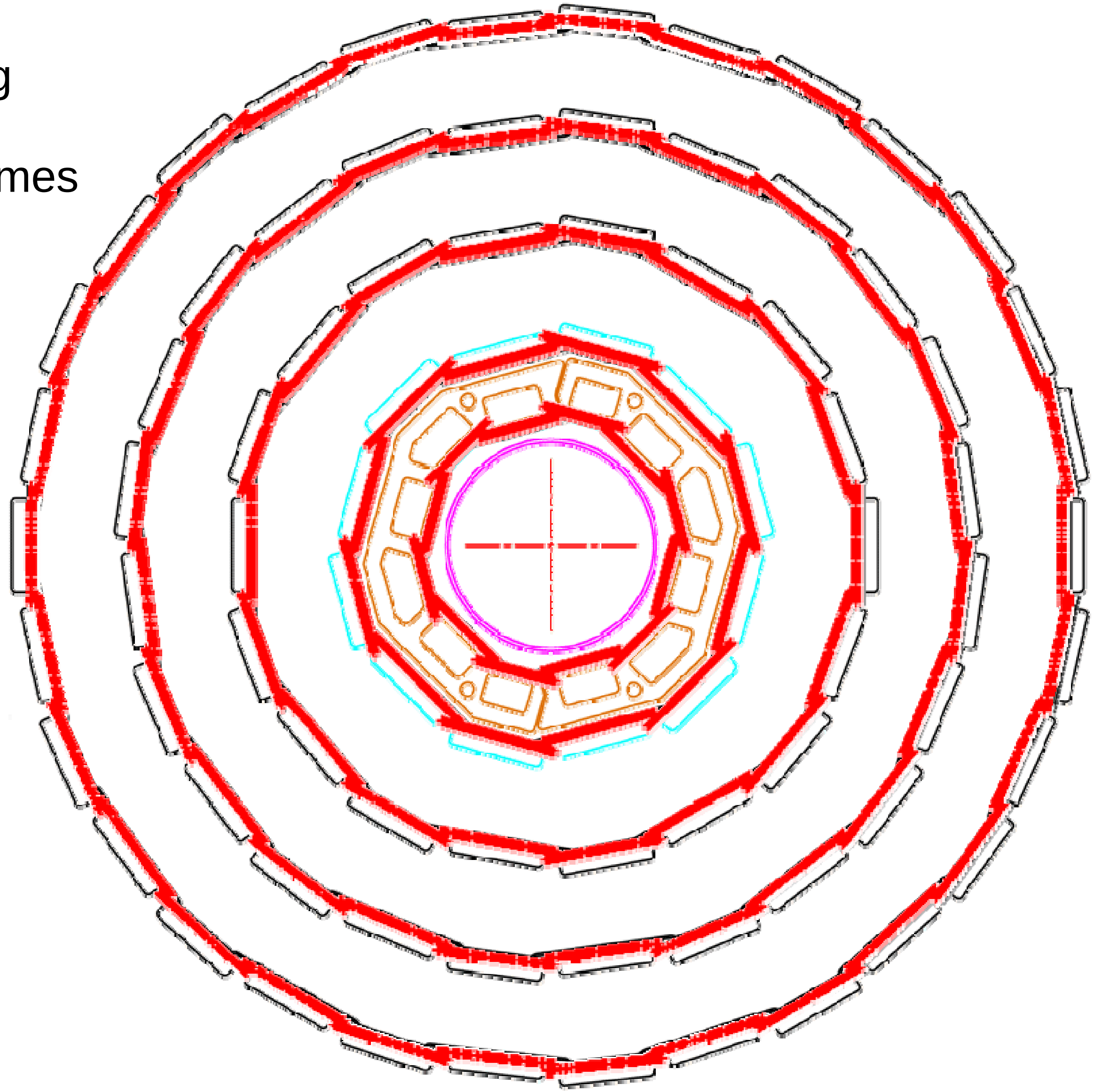


LCIO SimTracker Hits from Vertex

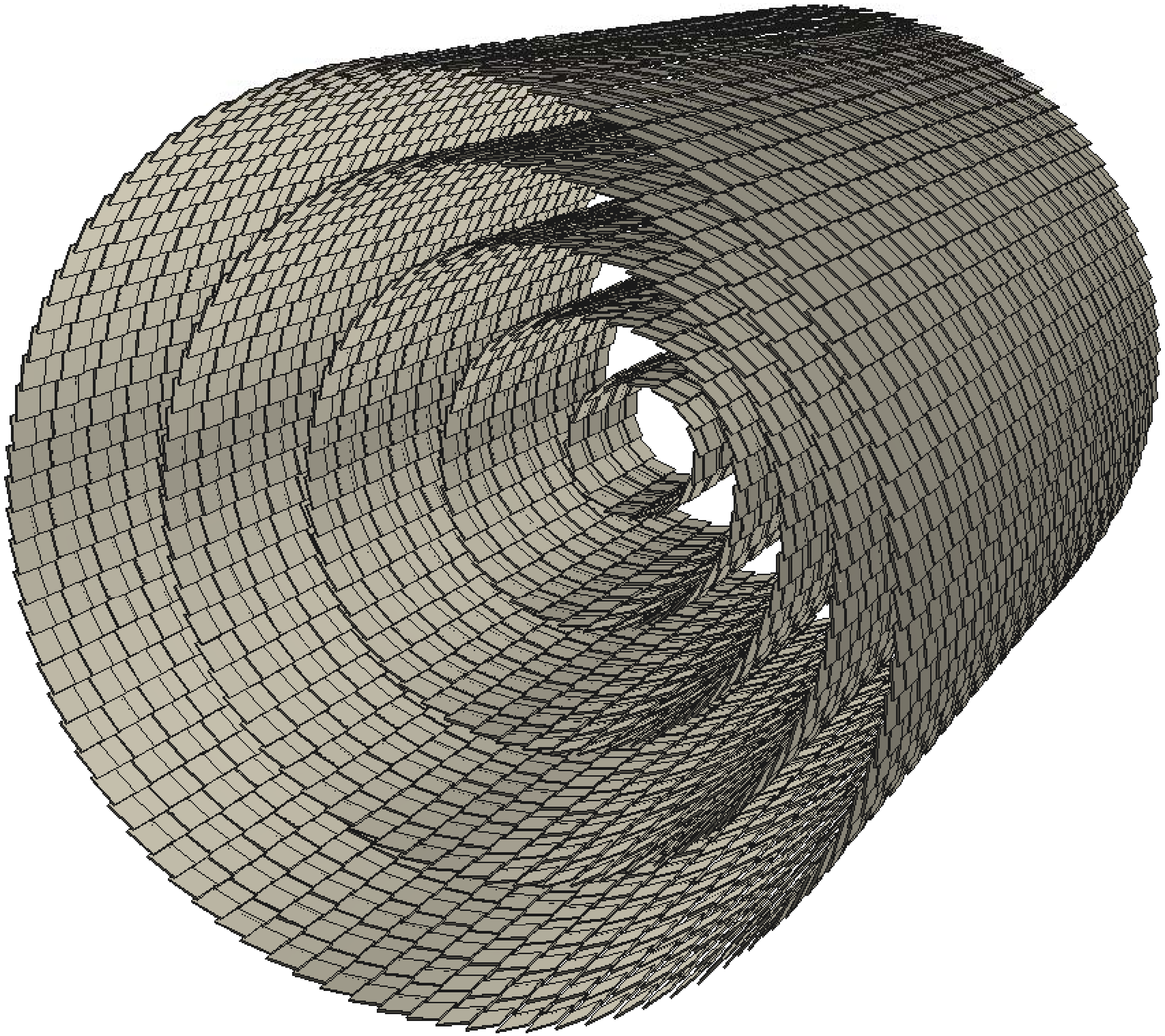
CAD Drawing

GEANT Volumes

LCIO Hits

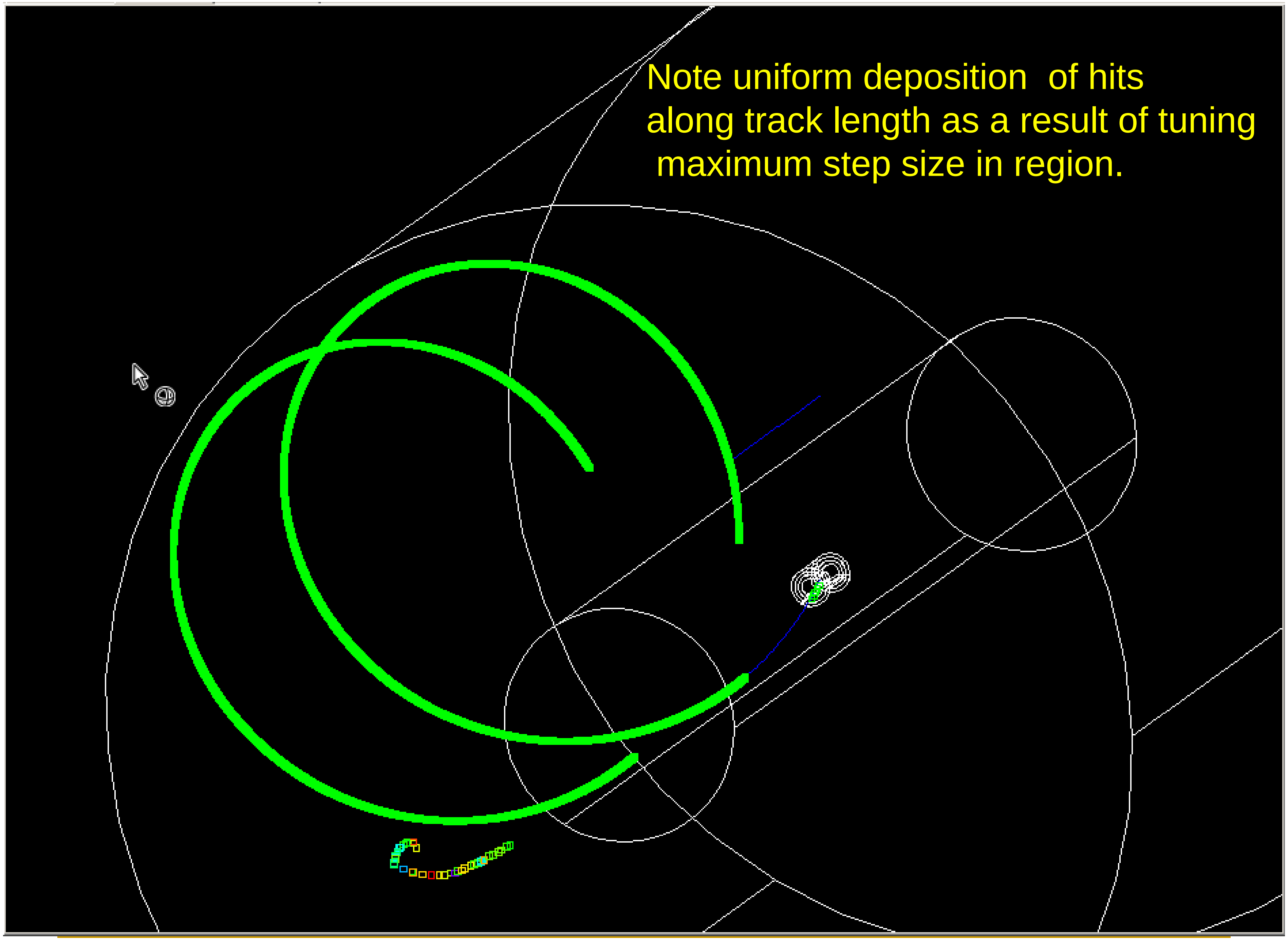


The Barrel Outer Tracker



TPC Simulations

- Most simulations to-date have created single hit at intersection with pad row “cylinder”.
- Not too bad an approximation for stiff tracks, but causes problems for loopers.
- Can improve simulations with a combination of range cuts and maximum step size cuts.
- These are configurable by region (themselves configurable) in the compact description.
  - Can define them differently for silicon and TPC.
  - Can change them at runtime to study settings.

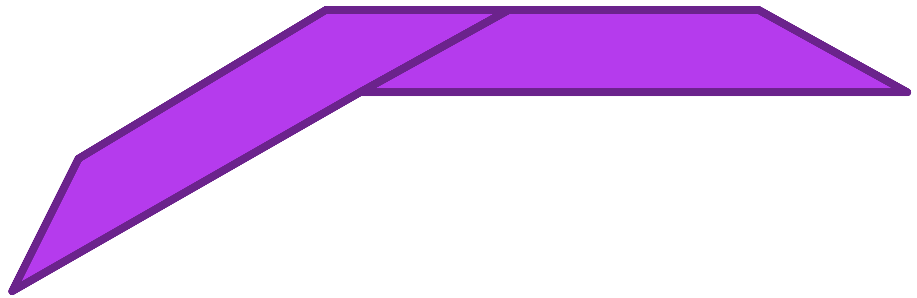

Note uniform deposition of hits
along track length as a result of tuning
maximum step size in region.

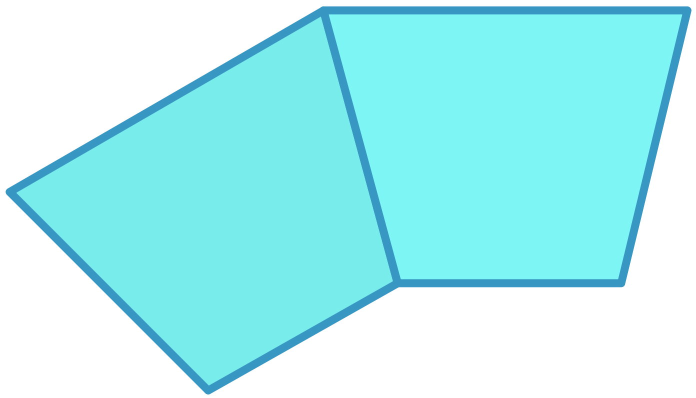
Calorimeter Improved Simulations

- Having settled on a concept with the requisite performance, will have to design a detector which can be built.
- Engineering will have to be done to come up with the plans, but the existing simulation package can already handle arbitrarily complex shapes.
- Can then study effects of support material, dead regions due to stay-clears, readout, power supplies, etc.
- However, hard work is in analyzing this, not simulating it.

Improved Calorimeter Simulations II

- Have two types of polygonal barrel geometries defined in the compact description:
- Overlapping staves: Wedge staves:





- Can define ~arbitrary layerings within these envelopes to simulate sampling calorimeters.

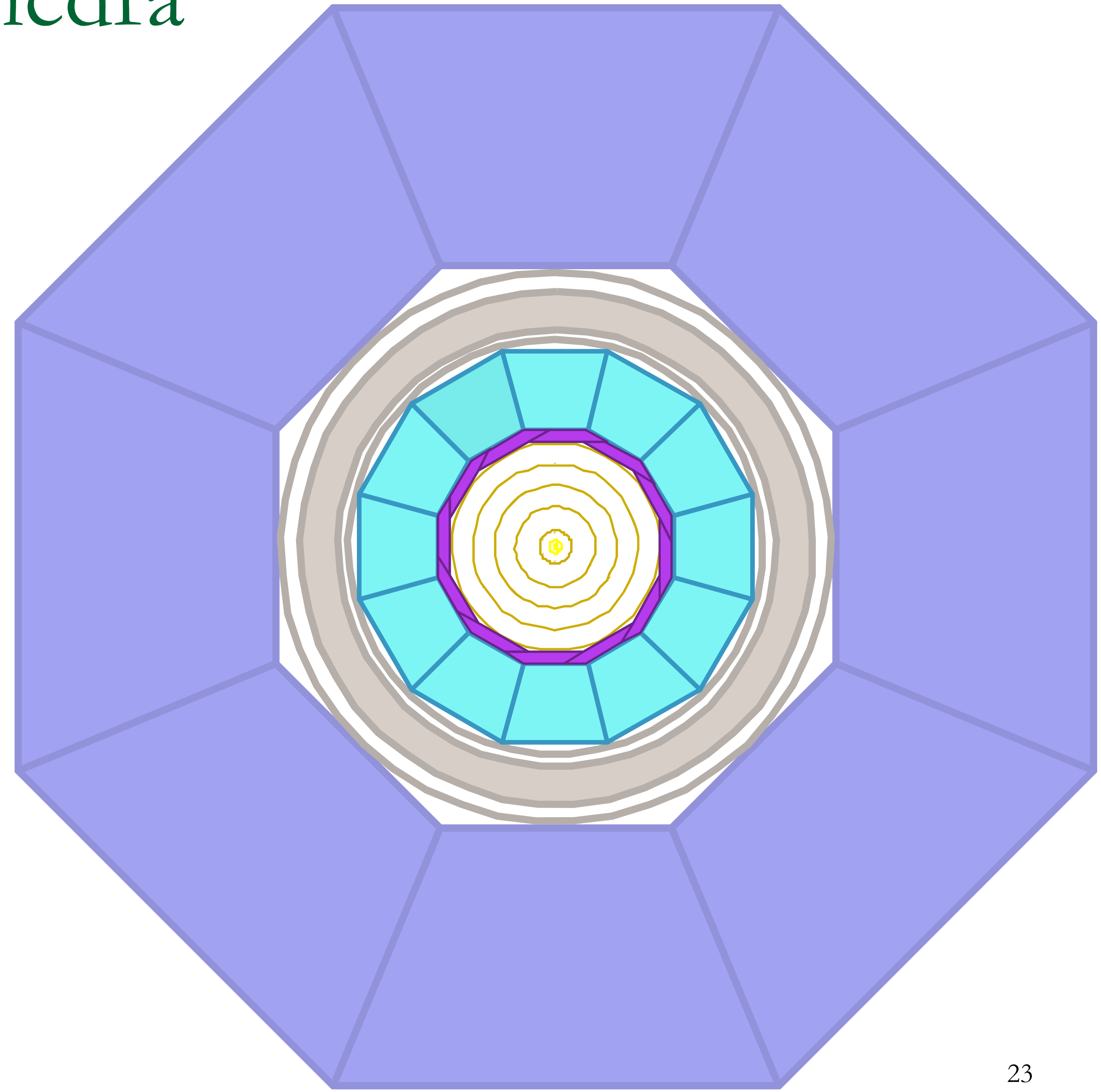
sid01_polyhedra

Dodecagonal, overlapping stave EMCal

Dodecagonal, wedge HCal

Cylindrical Solenoid with substructure

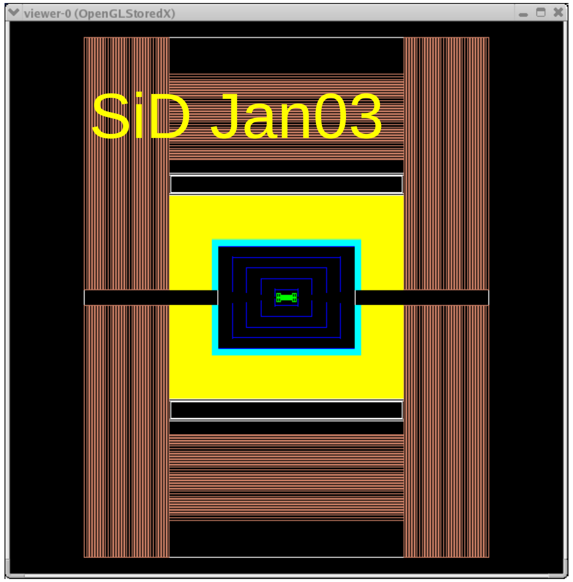
Octagonal, wedge Muon

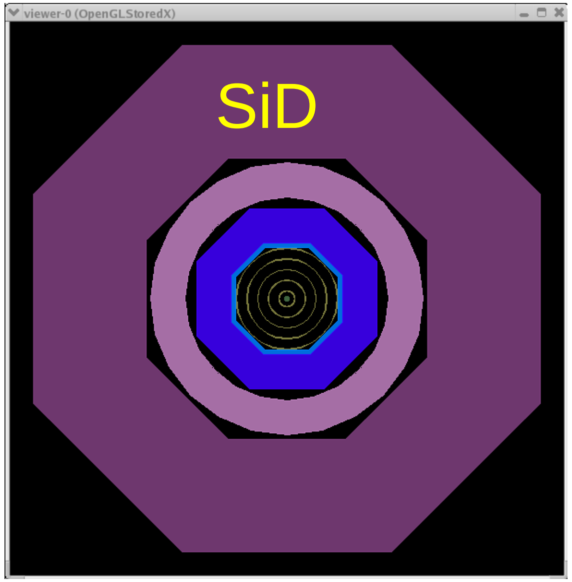


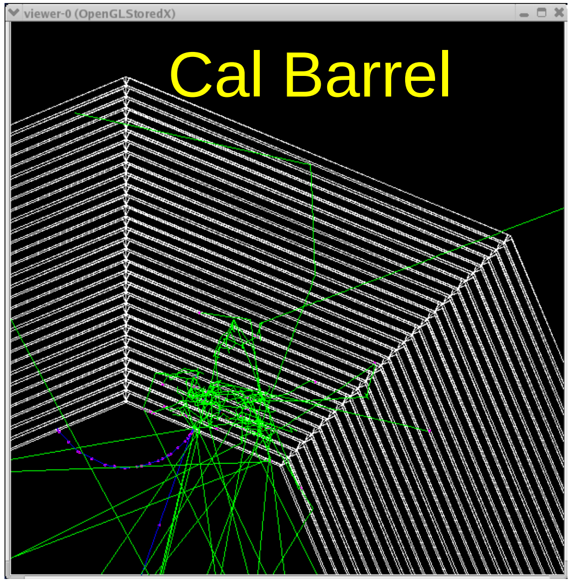
Detector Variants

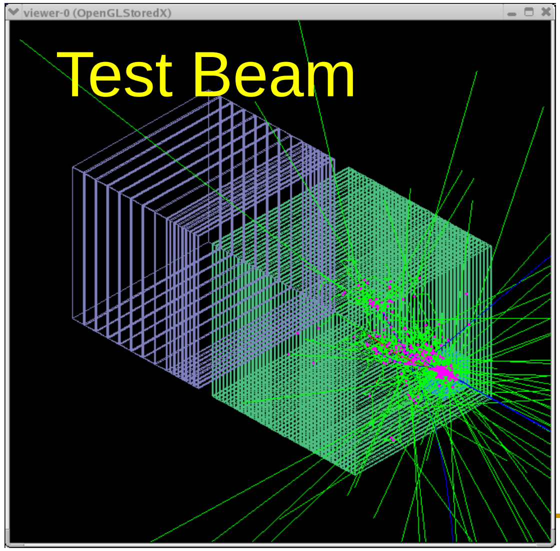
- Runtime XML format allows variations in detector geometries to be easily set up and studied:
  - Stainless Steel vs. Tungsten HCal sampling material
  - RPC vs. GEM vs. Scintillator readout
  - Layering (radii, number, composition)
  - Readout segmentation (size, projective vs. nonprojective)
  - Tracking detector technologies & topologies
    - TPC, Silicon microstrip, SIT, SET
    - “Wedding Cake” Nested Tracker vs. Barrel + Cap
  - Field strength
  - Far forward MDI variants (0, 2, 14, 20 mr)

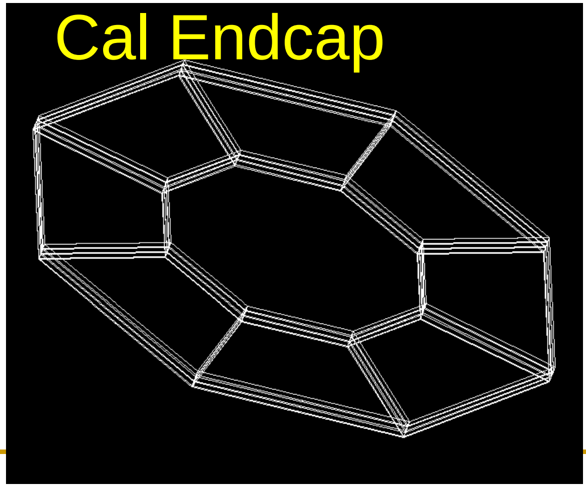
Example Geometries



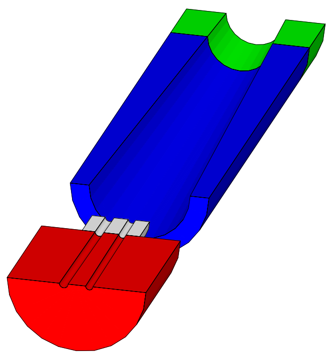




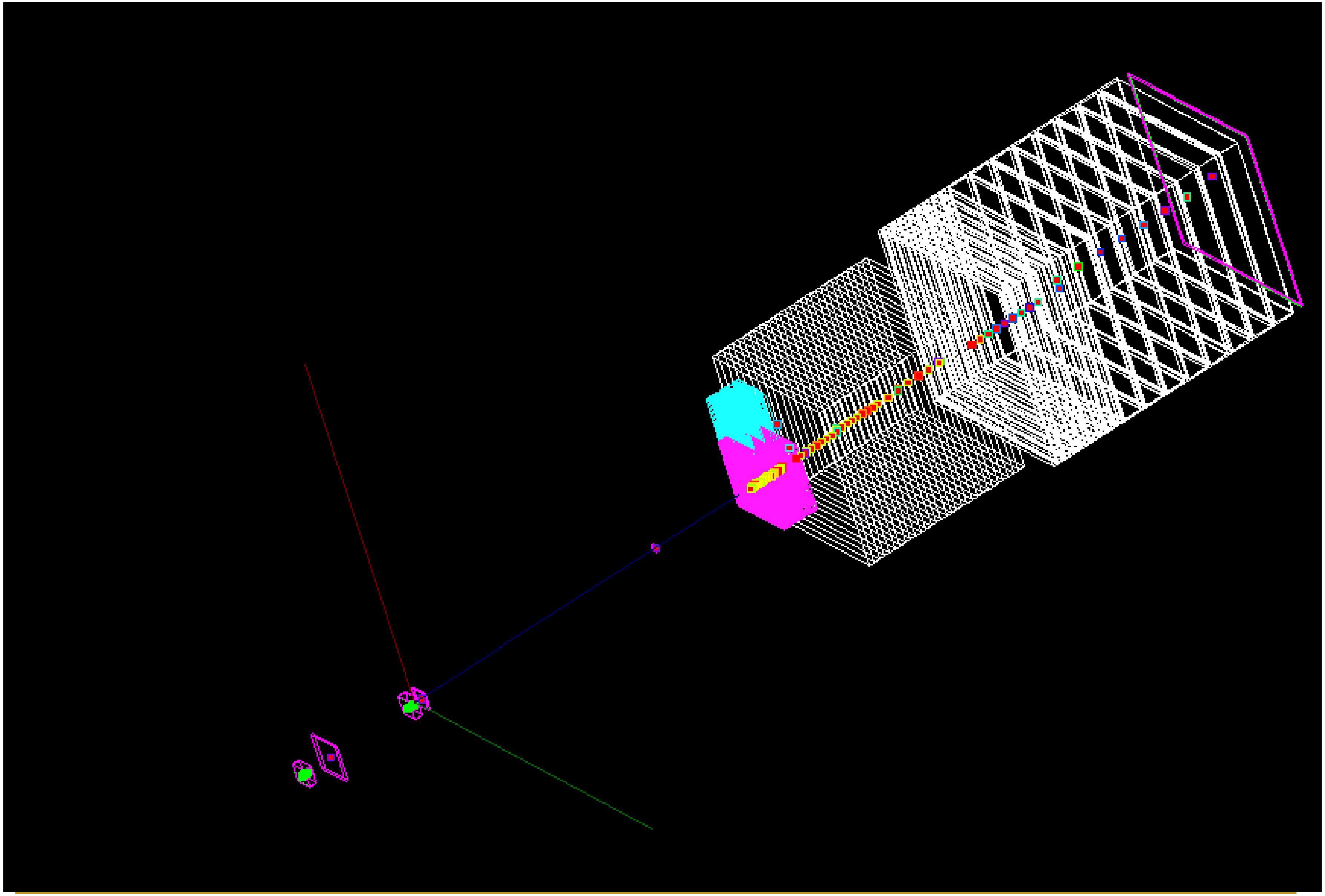




MDI-BDS



Example of Test Beam Analysis



slic

- Number of internal optimizations and refactorings.
  - Should not be noticed by end users.
- Upgrades to recent version of Geant4 has essentially eliminated problem of event aborts when particle tracking became stuck.
- slic from scratch:

```
cvs -d :pserver:anonymous@cvs.freehep.org:/cvs/lcd co SimDist
cd SimDist
./configure
make
```

- Binaries also available for Windows, Mac, Linuxes

“Signal” and Diagnostic Samples

- Have generated canonical data samples and have processed them through full detector simulations.
- simple single particles: γ, μ, e, $\pi^{+/-}$, n, …
- composite single particles: π^0,ρ, K^0_S ,τ, ψ, Z, …
- Z Pole events: comparison to SLD/LEP
- WW, ZZ, tt, qq, tau pairs, mu pairs, Zγ, Zh:
- Web accessible:

http://www.lcsim.org/datasets/ftp.html

Backgrounds

- Cain (to be done) & GuineaPig pairs and photons.
  - Add crossing angle, converted to stdhep
- Muons and other backgrounds from upstream collimators & converted to stdhep.
- $\gamma\gamma \rightarrow$ hadrons generated as part of the "2ab^{-1} SM sample."
- All events then capable of being processed through full detector simulation.
- Additive at the detector hit level, with time offsets, using LCIO utilities.

LCIO Utilities

- A number of LCIO file-handling tasks have been assembled and are available as command-line options.

  > lcio -h

  usage: LcioCommandLineTool

  Commands:

  compare

  concat

  validate

  siodump

  print

  stdhep

  split

  random

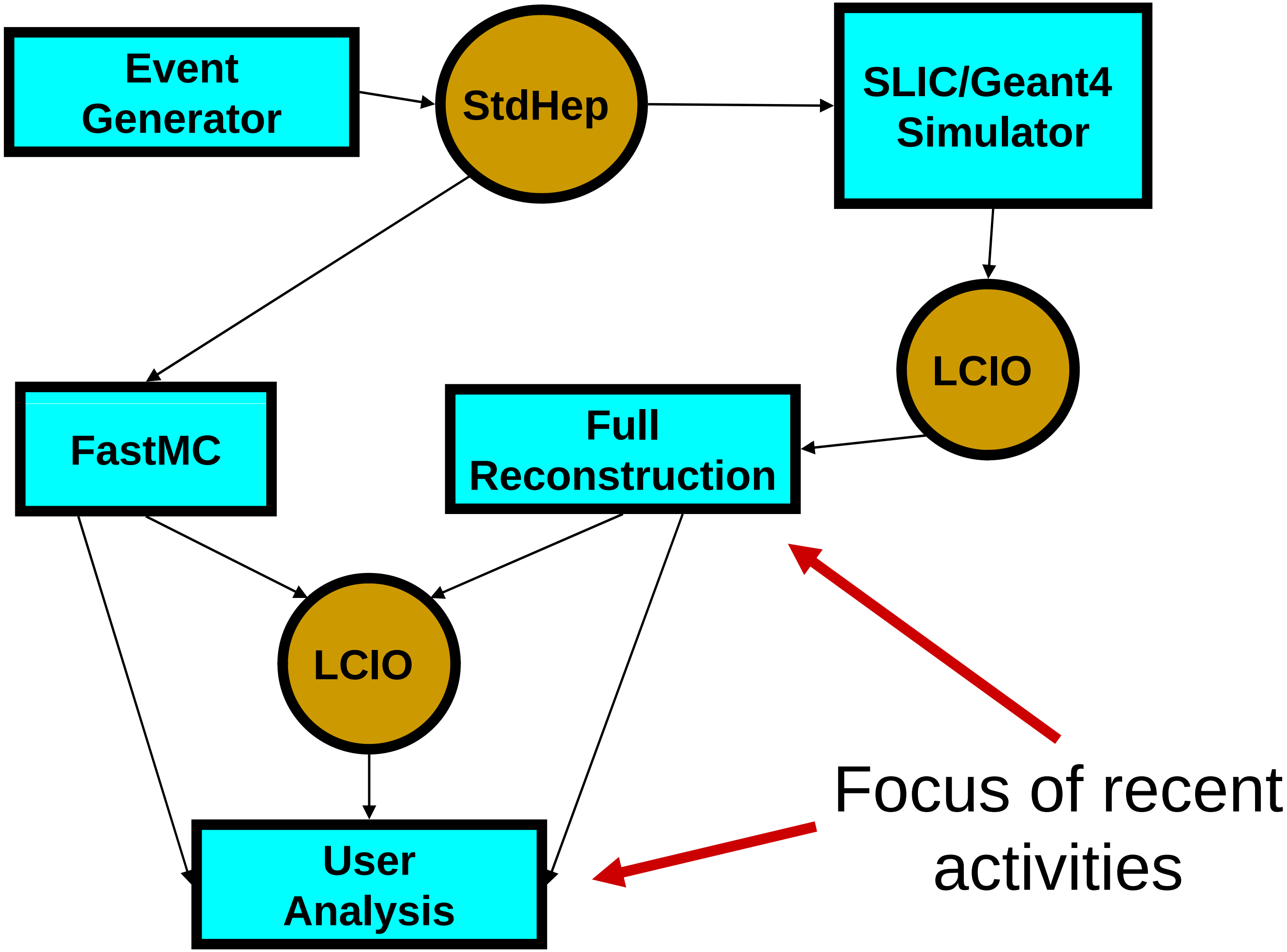
  count

  merge

  -h Print lcio command-line tool usage.

  -v Set the verbosity.

How to run full reconstruction?



org.lcsim Reconstruction Packages

Contrib

Package	Author	State	Docs/Talks	Description
org.lcsim.contrib.CalAnal	?	?		?
org.lcsim.contrib.CarstenHensel	Carsten Hensel	?		HMatrix cluster analysis
org.lcsim.contrib.Cassell.recon.Cheat	Ron Cassell	?		Cheat Recon driver
org.lcsim.contrib.EricBenavidez.EMClusterID	Eric Benevides	?		HMatrices analysis of single particle events
org.lcsim.contrib.JanStrube.tracking	Jan Strube	complete JUnit tests Javadoc	A New Track Interface	Alternate implementation of Track, FastMCTrack, Swimmer. Awaiting incorporation into main body of code
org.lcsim.contrib.JanStrube.vtxFitter	Jan Strube	incomplete		Vertex fitter, using the Kalman approach by Grab, Luchsinger. Add the VtxFitterDriver from the sandbox to get an idea of the current status
org.lcsim.contrib.JanStrube.zvtop	Jan Strube	incomplete		ZVTop implementation, taking advantage of the new Track interface, alpha quality
org.lcsim.contrib.KFFiter	Fred Blanc	?		Kalman Filter Fitter
org.lcsim.contrib.LeiXia	Lei Xia	?		PFA analysis
org.lcsim.contrib.NickSinev.tracking.wmfitter	Nick Sinev	?		SLD Weight matrix fitter
org.lcsim.contrib.NickSinev.ztracking	Mike Ronan+Nick Sinev?	?		Track cheater?
org.lcsim.contrib.onoprien.mcTrackFinder	D. Onoprienko	complete		Configurable cheater track finder and related utilities.
org.lcsim.contrib.onoprien.tester	D. Onoprienko	functional, under development		Track finder performance testing suite
org.lcsim.contrib.SODTracker	Fred Blanc	?		Silicon Outer Detector (SOD) Tracker
org.lcsim.contrib.SiStripSim	Tim Nelson	?		Silicon Strip Simulation (moving soon to org.lcsim.detector)
org.lcsim.contrib.SteveMagill	Steve Magill	?		PFA Analysis example
org.lcsim.contrib.niu	Vishnu and Guilherme	?		NIU PFA code
org.lcsim.contrib.proulx	?	?		?
org.lcsim.contrib.seedtracker	Richard Partridge	?		Tracking algorithm based on forming track seeds from all 3-hit combinations
org.lcsim.contrib.subdetector.tracker.silicon	Tim Nelson	?		Experimental geometry package (Developed further in Geomconverter as org.lcsim.detector by Jeremy)
org.lcsim.contrib.tracking	Tim Nelson	?		Outer-tracker-only track finding
org.lcsim.contrib.uiowa	Mat Charles	unstable		Template-style PFA implementation (NonTrivialPFA)

Production

Package	Author	State	Docs/Talks	Description
org.lcsim.digisim	Guilherme Lima	?		Calorimetry digitization simulator
org.lcsim.mc.CCDSim	Nick Sinev	?		CCD digitization
org.lcsim.mc.fast	Many	?		Fast MC package, including tracking, calorimetry
org.lcsim.recon.cat	D. Onoprienko E. von Toerne	functional, under development		Calorimeter Assisted Track Finder
org.lcsim.recon.cheater	Mike Ronan	?	confluence	Recon cheater
org.lcsim.recon.cluster.analysis	Ron Cassell	?		Generic cluster performance analysis
org.lcsim.recon.cluster.cheat	Ron Cassell	?		Cluster cheater
org.lcsim.recon.cluster.clumpfinder	Mat Charles	?		finds dense clumps within clusters
org.lcsim.recon.cluster.directedtree	G.Lima, J.McCormick, Vishnu	?		Directed tree cluster finder
org.lcsim.recon.cluster.fixedcone	Norman Graf	?		Cluster finder
org.lcsim.recon.cluster.mipfinder	Wolfgang Mader, Mat Charles	stable		MIP finding
org.lcsim.recon.cluster.mst	Mat Charles	stable		Minimal spanning tree cluster finder
org.lcsim.recon.cluster.nn	Norman Graf	?		Nearest neighbout cluster finder
org.lcsim.recon.cluster.structural	Mat Charles	stable		Specialized clusterer for hadronic showers
org.lcsim.recon.emid.hmatrix	Norm Graf	?		HMatrix package
org.lcsim.recon.ganging	Ron Cassell	?		Allows virtual ganging of calorimeter hits
org.lcsim.recon.muon	C. Milstene	?		Muon finding
org.lcsim.recon.particle	Ron Cassell	?		Perfect PFA
org.lcsim.recon.pfa.cheat	Mat Charles	functional		Cheating tools for PFA
org.lcsim.recon.pfa.identifier	Mat Charles	functional		Turn more primitive objects (clusters, tracks, etc) into ReconstructedParticles
org.lcsim.recon.pfa.output	Mat Charles	?		Modules to produce standard plots for PFAs
org.lcsim.recon.pfa.structural	Mat Charles	?	incomplete	Iowa PFA implementation (when stable) and associated tools
org.lcsim.recon.tracking.cheat	Ron Cassell	?		Track Cheater
org.lcsim.recon.tracking.ftf	?	?		?
org.lcsim.recon.tracking.trf	Norm Graf	?		TRF track finder + fitter
org.lcsim.recon.vertexing.billoir	Norman Graf, (Jan Strube)	incomplete		vertex fitting based on Billoir's method. Needs testing
org.lcsim.recon.vertexing.zvtop4	Jan Strube	incomplete		Vertex finding/fitting, awaiting completion of a vertex fitter
org.lcsim.recon.ztracking	M. Ronan	?		Track cheater

- Conclusions
  - Many people are working on reconstruction code
  - Effort to persuade people to commit code to “contrib” area has been successful
  - But it is not easy for new users to understand how to use or contribute
- Working to extend tutorials to also cover reconstruction packages
  - Encourage developers to contribute documentation
  - Start by updating: http://confluence.slac.stanford.edu/x/f3c
  - Need more realistic analysis examples (help from benchmarking and physics groups?)
  - Extend PFA template idea to full reconstruction

Reconstruction

- Many of the core reconstruction algorithms (track finding, fitting, calorimeter clustering, etc.) are in place.
- Have defined interfaces for a number of tasks, with many different plug-&-play implementations (e.g. calorimeter clustering).
- Standardized algorithm comparison tools.
- Standard calorimeter calibration procedures.
- Concentrating on implementing a template for individual particle reconstruction:
  - Decouples interdependencies of different tasks.
  - Allows comparisons between different algorithms or implementations.
  - Easily swap in MC “cheater” to study effects of particular analysis task, independent of other tasks.

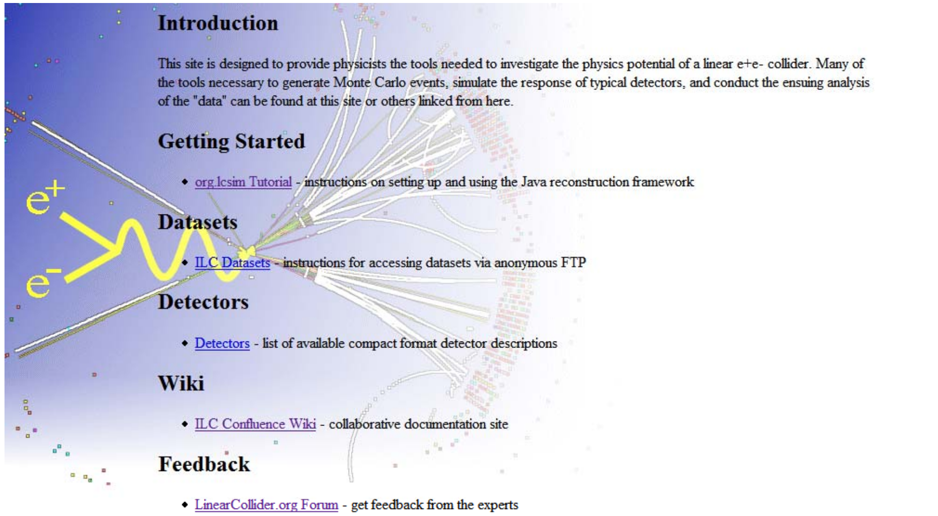
The Grid

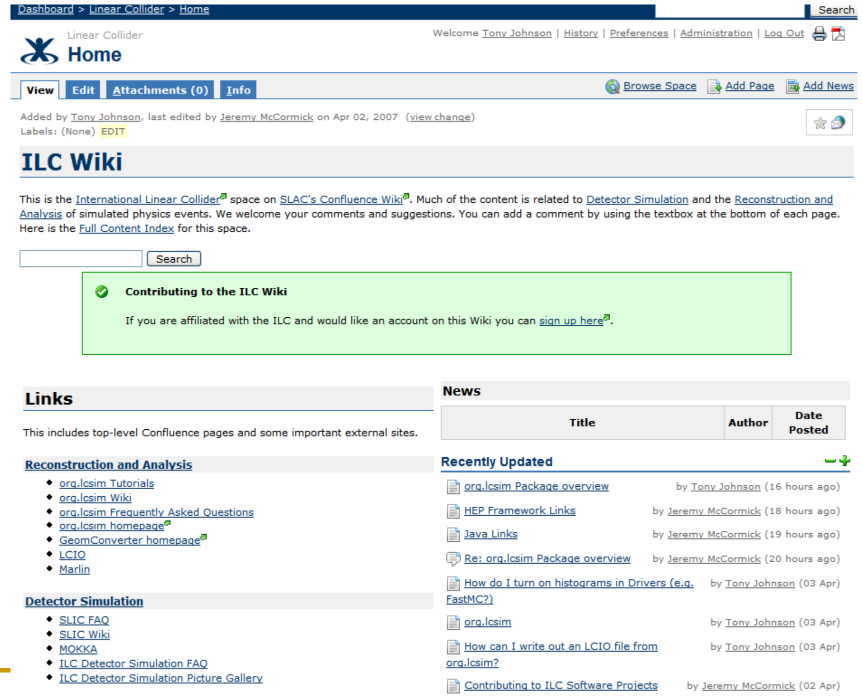
- Existing resources have proven sufficient to-date for event generation, detector response simulation and reconstruction/analysis.
- Grid tools seem to be getting to the point where they are useful, so are beginning transition.
- Tools have been developed from the beginning to be grid friendly, i.e. static binaries, no db connections, …
- Have developed (SBIR w/ Tech-X) Interactive Dataset Analysis on the Grid tools (as opposed to normal batch processing).
  - Plug-in allows grid analysis from within JAS.

Resources for getting started

- http://lcsim.org/ Web Site
  - Tutorials
    - Software installation
    - Using tools
    - Simple Analysis Examples
    - Developers Guide
  - Datasets
  - Documentation

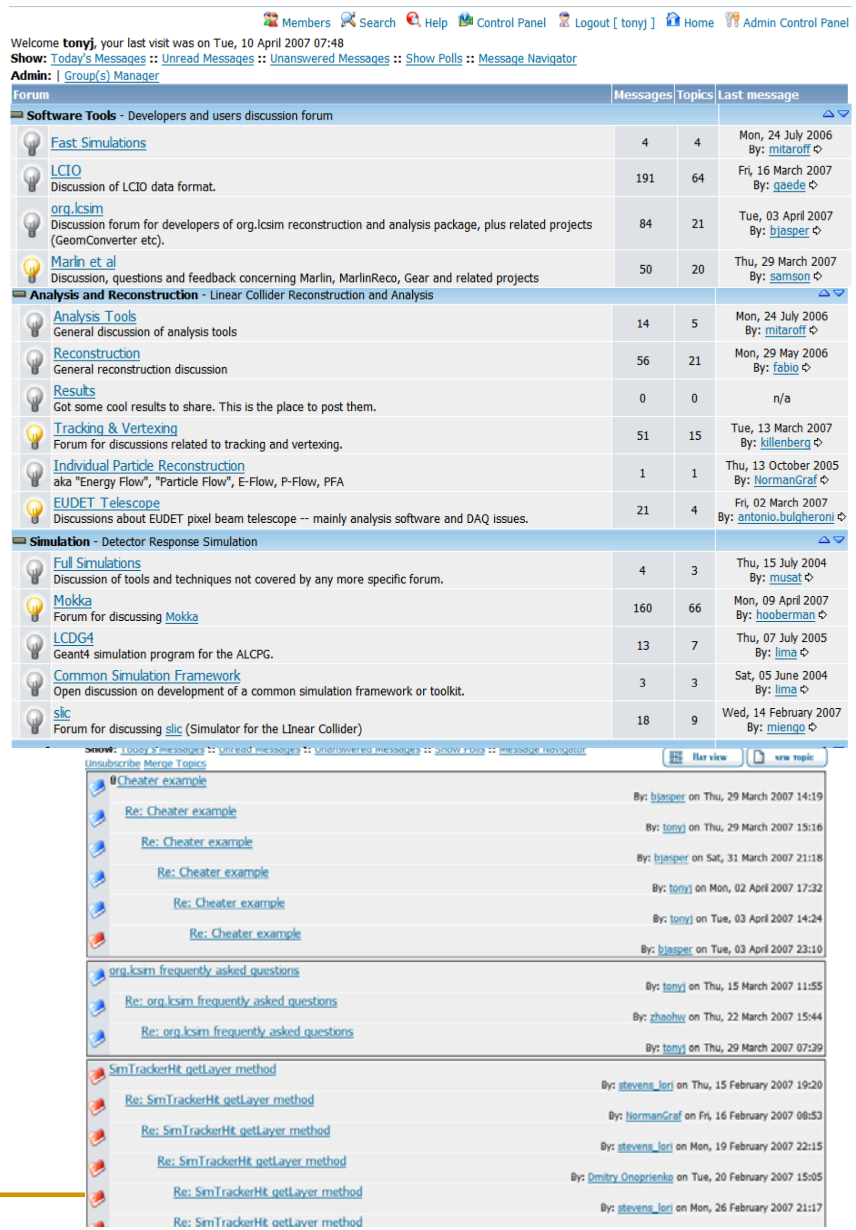
- Confluence Wiki
  - More tutorials
  - More documentation
  - Frequently asked Questions
  - You are encouraged to comment on, add to, or correct existing documentation
    - https://jira.slac.stanford.edu/signup





Resources for getting started

- Discussion Forums
  - http://forum.linearcollider.org/
    - SLIC, org.lcsim
  - Not recommended
    - Spray E-mail to developers
      - Banging head against wall
      - Uninstall and reinstall software 3 times
  - Recommended
    - Post questions on the forum
      - You will get faster answers
      - You will get more accurate answers
      - Others will benefit from seeing answers to your questions
    - Discuss what you would like to do
      - get feedback on best practices



ALCPG Simulation Summary

- ALCPG Sim/Reco team supports an ambitious detector simulation effort.
- Goal is flexibility and interoperability, not technology or concept limited.
- Provides full data samples for ILC physics studies.
  - Stdhep and LCIO files available on the web.
- Provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description.
- Reconstruction & analysis framework exists, core functionality available, individual particle reconstruction template developed, various analysis algorithms implemented.
- Need to iterate and apply to various detector designs.

Additional Information

- lcsim.org - http://www.lcsim.org
- ILC Forum - http://forum.linearcollider.org

- Wiki - http://confluence.slac.stanford.edu/display/ilc/Home
- org.lcsim - http://www.lcsim.org/software/lcsim
- Software Index - http://www.lcsim.org/software
- Detectors - http://www.lcsim.org/detectors

- 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