lcsim: Status and Future

Norman Graf, Jeremy McCormick
(SLAC)
LCWS13, Tokyo
11/12/13

Common LCD Software

- General agreement among LC concept working groups that most efficient way forward is to move to more common software tools
- Already have an unprecendently strong base of collaborative software used by CLIC, ILD and SID:
    - LCIO - common EDM and persistency
    - PandoraPFA, LCFIVertex/LCFIPlus
    - Geant4 (slic, Mokka)
- Informal series of Linear Collider Software Meetings held at CERN with software experts from CLIC, ILD and SiD in 2009, 2012 & 2013
    – Continue to identify areas for collaborative development.

LCD Software Meetings

- At 2012 meeting, reached a consensus to work towards:
  - a common simulation application based on the geometry description developed in AIDA WP2
  - a common C++ tracking package in the context of AIDA WP2
- At 2013 meeting discussion focused on the details of how these goals can be achieved
  - interface between geometry description and simulation
  - interface to reconstruction (tracking)
  - agreement to use DD4hep as geometry tool
  - decided to develop prototypes to investigate options

MC Events

- MC Event Generation
  - Whizard talk at recent SiD workshop by Jürgen Reuter
  - Looking forward to inclusion of lumi-linker files and other improvements and modifications made by Tim Barklow during LOI/DBD exercises
  - Looking forward to direct LCIO output
- MC Hierarchy handling in slic
  - Handling of secondaries, simulator status codes, etc. has grown organically over a decade, complicated
  - Would like to refactor, with input from Mokka developers/users, to provide expected behavior.

Full Detector Response Simulation

- Use Geant4 toolkit to describe interaction of particles with matter and fields.
- Program framework provides access to:
  – Event Generator particle input
  – Detector Geometry description input
  – Detector Hits output
- slic used primarily by SiD
- Mokka used primarily by ILD
  – support at LLR has been reduced
  – main developers and maintainers moved on to other tasks

Geometry Definition

- Goal is 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.

LCDD and GDML

- Adopted GDML as base geometry definition, then extended it to incorporate missing detector elements.

LCDD

- detector info
- identifiers
- sensitive detectors
- regions
- physics limits & cuts
- visualization
- magnetic fields

GDML

- expressions (CLHEP)
- materials
- solids
- volume definitions
- geometry hierarchy

LCDD Structure

<lcdd>	LCDD Root Element
<header>	Information about the Detector
<iddict>	Identifier Specifications
<sensitive_detectors>	Detector Readouts
<limits>	Physics Limits
<regions>	Regions (sets of volumes)
<display>	Visualization Attributes
<gdml>	GDML Root Element
<define>	Constants, Positions, Rotations
<materials>	Material Definitions
<solids>	Solid Definitions
<structure>	Volume Hierarchy
</gdml>	
<fields>	Magnetic Field
</lcdd>	

lcdd Features

- **Regions**: production cuts
- **Physics limi**ts: track length, step length, etc.
- **Visualization**: color, level of detail, wireframe/solid
- **Sensitive detectors**
  - calorimeter, optical calorimeter, tracker
  - segmentation
- **ID**s
  - volume identifiers (physical volume id)
- **Magnetic field**s
  - dipole, solenoid, field map
- utilities
  - information on Geant4 stores
  - GDML load/dump

lcdd recent changes

- added doc including doxygen tags to all functions & classes
- reorganized sources into separate directories
- extracted sensitive detector ProcessHits() code into separate HitProcessor classes
- removed optical_calorimeter and other sensitive detector classes which were replaced by HitProcessors
- added examples to lcdd/examples (still in progress)
- added classes CellReadout, CellReadout2D, and CellReadout2DSegmentation as prototypes of possible refactoring for Segmentation-based classes
- http://www.lcsim.org/software/lcdd/doc/html/
- http://www.lcsim.org/software/lcdd/doc/html/annotated.html

"Compact" Description

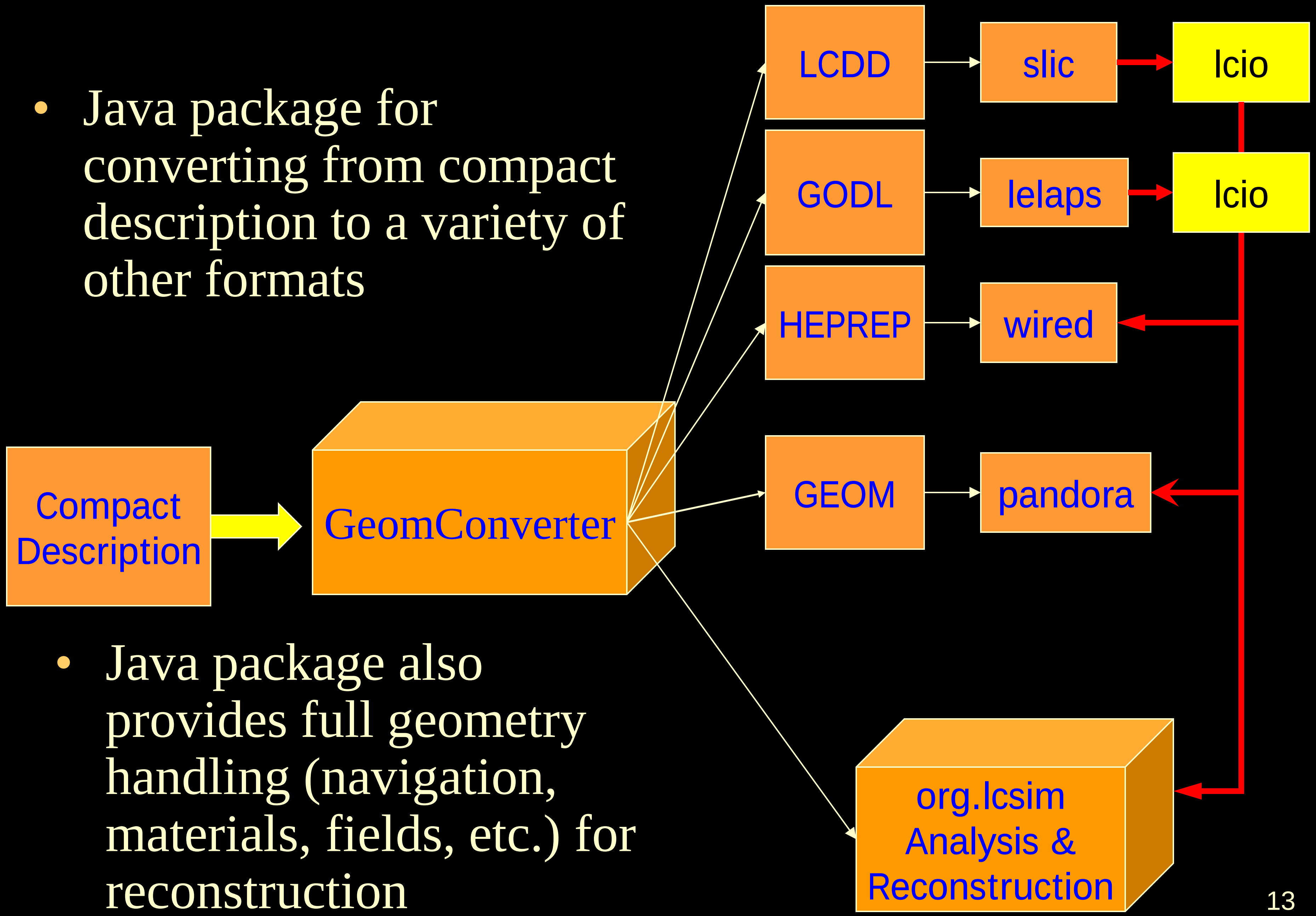
- The lcdd file is very descriptive, but therefore also very verbose.
- Can be written by hand, but prone to human error.
  - Also, just specific to the simulation and not easily accessible to reconstruction and visualization.
- Developed a "compact" detector description which encapsulates the basic properties of a detector and which is further processed by code to produce the input specific to different clients.

Compact Detector Description

- A number of generally useful detector types (at least for HEP collider detectors) have been developed, such as:
  - Sampling calorimeters
  - TPCs
  - Silicon trackers (microstrip as well as pixel)
  - Generic geometrical support structures
- Can also incorporate GDML snippets
  - Allows inclusion of more complicated volumes derived for instance from engineering (CAD) drawings.

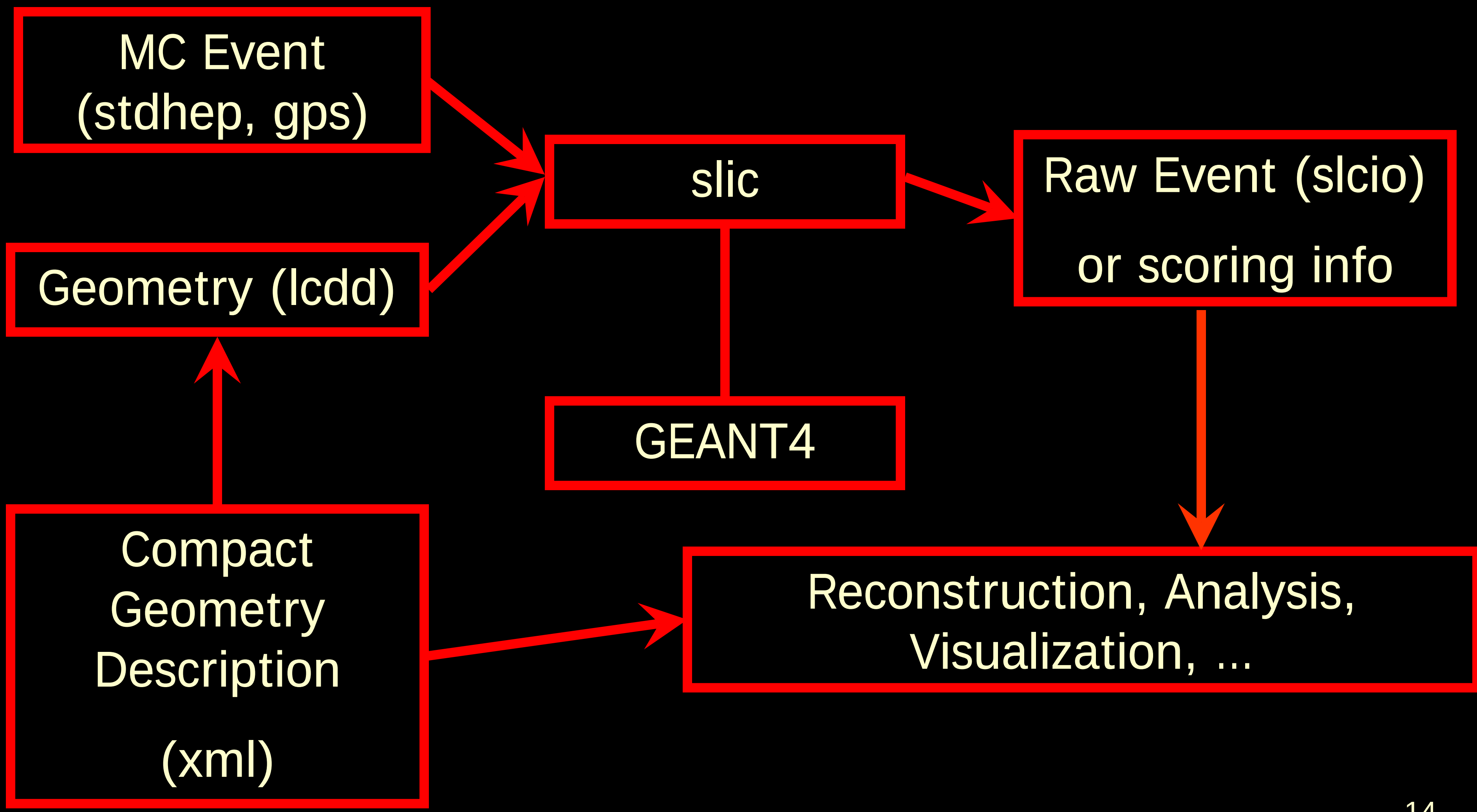
GeomConverter

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



- Java package also provides full geometry handling (navigation, materials, fields, etc.) for reconstruction

Detector Full Simulation



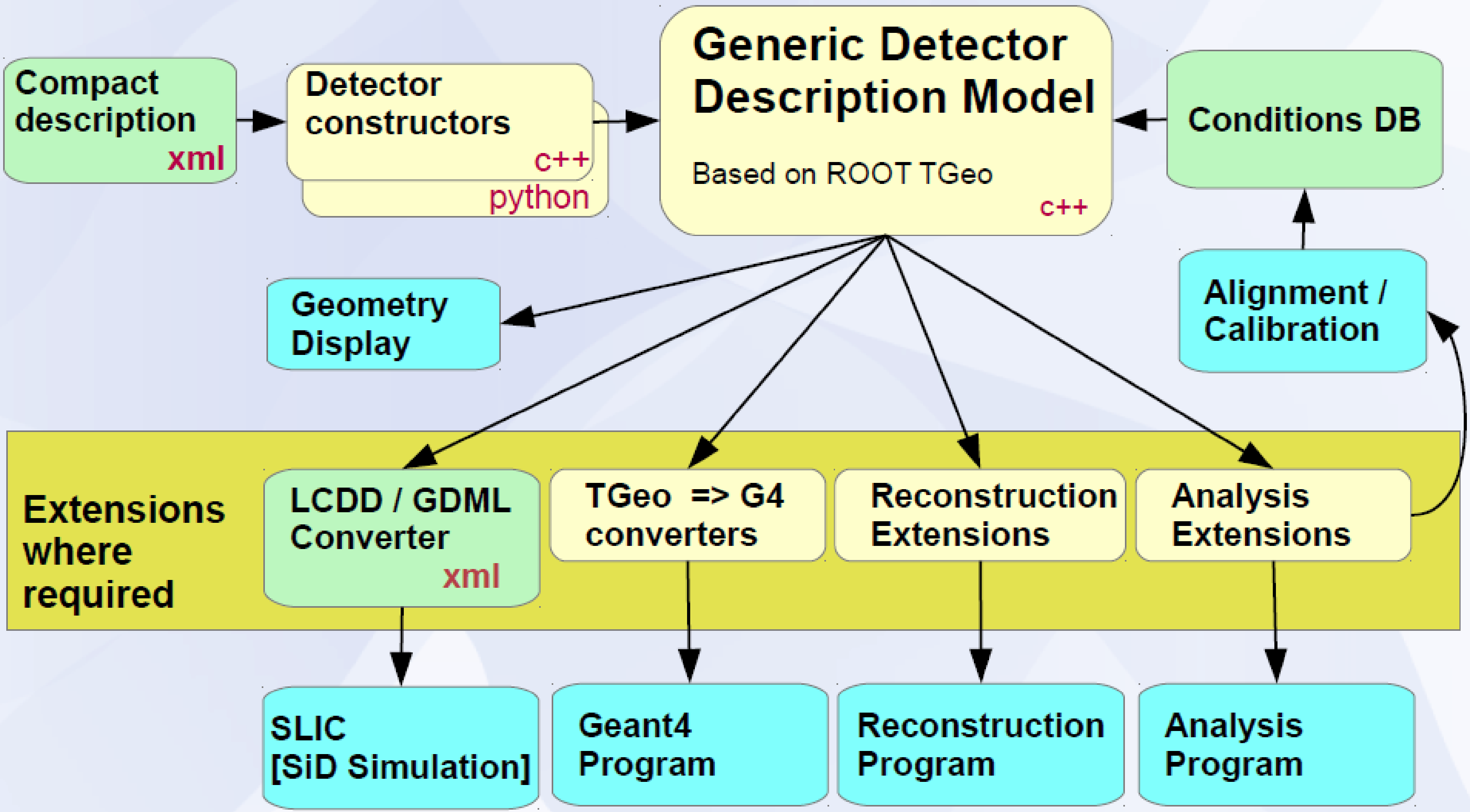
DD4hep

- AIDA deliverable to provide Detector Description for HEP experiments supporting the full experimental life cycle:
  - Detector concept development & optimization
  - Detector construction and operation
  - Simulation, reconstruction, analysis
  - Support for:
    - Geometry
    - Readout
    - Alignment
    - Calibration

DD4hep and slic

- Instead of reinventing another Geant4 program, will use slic as the simulation program.
- Instead of reinventing a geometry manipulation & navigation program, will use root's TGeo class.
- Will remain decoupled from slic code base by producing lcdd file.
- Virtual segmentation classes (used e.g. for calorimeter readout) will be shared between simulation and C++ reconstruction.

DD4hep Overview



"Virtual Segmentation"

- Unrealistic to include all pixels in LC detectors as physical or logical elements in Geant4
- Unrealistic to store every step of every particle (especially secondaries in calorimeter showers)
- Use "virtual segmentation" to aggregate energy depositions in calorimeters.
  - Fixed number of different types
  - position to cell ID (used in simulation)
  - cell ID to position (used in reconstruction)
  - list of neighboring cell IDs (reconstruction)

Virtual Segmentation

- Working to provide package which provides implementations of basic types of segmentation.
- Want minimal coupling between simulation (Geant4) and reconstruction (root) environments.
- Refactoring of Geomconverter / lcdd packages.
- Ongoing work with Christian Grefe to explore calorimeter reconstruction and Frank Gaede to investigate track reconstruction using new API.
- Developing package to provide “standard” segmentations used in HEP detectors

DD4hep Status & Plans

- Have developed simple prototypes:
  - ILD: VXD, SIT, TPC, AHcal
  - Calice test beam
  - CLICSiD
- Being used to study technical issues:
  - cellIDs, detector segmentations
  - sensitive detectors
  - interface to reconstruction
- Working meeting planned for December @ CERN

ILCSoft

- System to build and release software packages
- Working towards using this for all common packages
- slic, XercesC, HepPDT, GDML, LCDD and slicPandora are now included
  - SimDist no longer being used
- DD4hep also included

trf improvements

- Improvements made to correctly handle inhomogeneous fields
  - 3D field map on regular Cartesian Grid now fully supported
  - Would like to collaborate with others to improve the interpolation scheme (both in trf and in slic)
- Improved handling of material effects (dE/dx in particular)
- Small misalignments handled as perturbations on ideal surfaces

org.lcsim : Tracking

- Primarily working on improvements required by HPS experiment to handle:
- non-ideal geometries and tracker alignment
- realistic field maps
- more refined material effects
  - e.g. low energy electrons
- track fitting and extrapolations in misaligned geometries and inhomogeneous fields.

org.lcsim : Calorimetry

- Modifications and improvements to support the SiD Ecal testbeam:
  - Use of hexagonal cells in both simulation and reconstruction
  - Robust clustering algorithms
  - Robust handling of dead & noisy channels
  - all the usual real-life problems encountered in testbeams

org.lcsim : Conditions

- Again, driven more by HPS and SiD Ecal testbeam
- More robust conditions database handling

Physics & Detector Studies

- lcsim toolkit was used in the generation of event samples used for the Snowmass exercise
  - primarily the Higgs working group using DBD SiD
- Currently planning a systematic round of detector optimization studies for SiD to study cost/benefit of different detector technologies and layouts
  - Tracker: Pixel vs strips, number and position of layers
  - Cal: Analog vs Digital Hcal, optimize Ecal
  - Overall: study overall dimensions and aspect ratio
- Collaboration welcomed!

Future Plans

- LCD sim/reco working groups continue to work towards increased software commonality.
- Midterm goal: all concept working groups use slic for full detector response simulation via lcdd
- Provide common source of geometry for simulation and (C++/root) reconstruction via DD4hep (TGeo)
- Work towards common reconstruction software where possible.