Software Status

Norman Graf
(SLAC)
SiD Meeting, SLAC
01/12/15

Introduction

- No major changes since ALCWS14, LCWS14.
- Continue to improve/refactor existing core software
  - slic (Geant4-based simulation program, C++)
  - org.lcsim (Java-based reconstruction, Java)
- Supporting individual analysis groups
  - Ecal testbeam
  - Fcal optimization
  - Detector occupancy studies
  - Pixel tracker resolutions
- Maintaining existing functionality as best we can

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.

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 encapsulates ALL of the information needed to run a Geant4 simulation.

lcdd recent changes

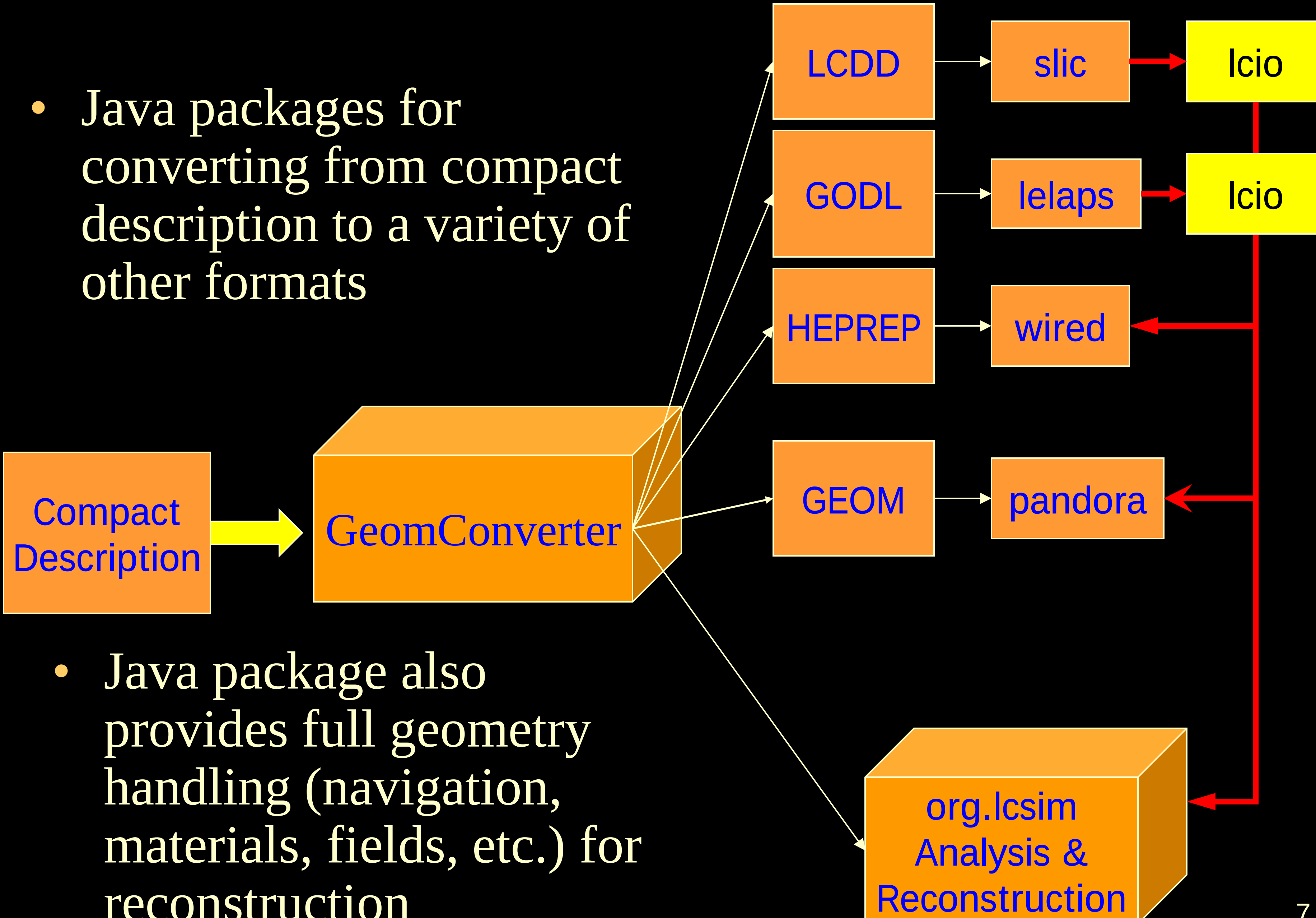
- added doc including doxygen tags to all functions & classes
- reorganized sources into separate directories
- extracted sensitive detector ProcessHits() code into separate HitProcessor classes
- removed optical_calorimeter and other sensitive detector classes which were replaced by HitProcessors
- 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
- Article submitted to journal for publication.

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

"GeomConverter"

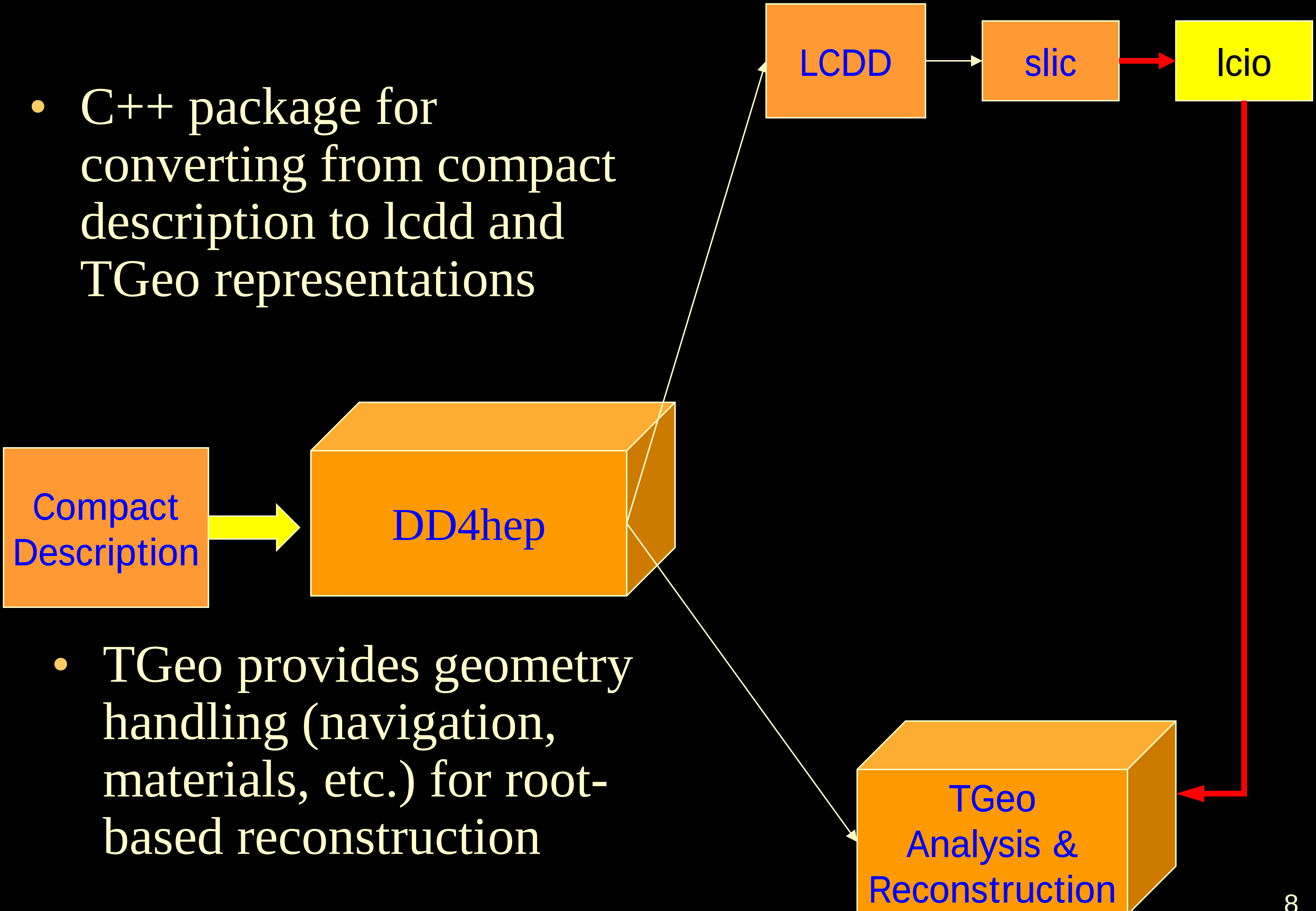
- Java packages for converting from compact description to a variety of other formats



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

DD4hep

- C++ package for converting from compact description to lcdd and TGeo representations



- TGeo provides geometry handling (navigation, materials, etc.) for root-based reconstruction

DD4hep and org.lcsim

- Reconstruction using the DD4hep software is envisioned to use the TGeo geometry navigation system. Only supports a root (C++) environment.
- Would have to make sure that:
  - the geometry navigation system used in the org.lcsim reconstruction packages continues to provide necessary functionality
  - the segmentation utilities continue to provide correct indexing and position information.

DD4hep and slic

- Had expected that with the end of support for Mokka that the LC community would adopt slic as its full-detector simulation program.
- DD4hep would write out LCDD as input to slic and provide TGeo support for root-based reconstruction. org.lcsim would continue to use compact.xml, but benefit from any common geometry development.
- But AIDA effort is being devoted to writing a new, from-scratch, Geant4-based simulation toolkit.
  – End-users code their own subdetector geometries, build their own executables.

Magnetic Fields

- Most studies to-date for LC have used a very simplified constant magnetic field along the z axis.
- Currently have support for:
  - 3D field map on regular Cartesian Grid using linear interpolation
  - Polynomial fit to 1D field (currently used for anti-DID)
  - Symmetric solenoidal field (Bz, Br)
  - Work ongoing to extend to Bspline and Bezier fits
- Slows down the simulation (need to study cost/benefit)
- Would like to benefit from others' optimization experience (esp. polynomial field fits)

3D Magnetic Field Map Definition

- Input (B_X,B_Y,B_Z) @ (x,y,z) on a regular Cartesian Grid
- Use linear interpolation to determine field at an arbitrary point within a box represented by surrounding grid points.
- Both memory and compute intensive, but simple and straightforward.
  - Would prefer to have polynomial fit to the field.
- Classes written for use in slic (C++) and lcsim (Java).
- org.lcsim code available to incorporate into lcdd and lcsim.
- Field map is defined in magnet coordinates.
  - Able to translate field in compact description
- Compact.xml usage:

```
<field
    type="FieldMap3D“
    name=“HPSDipoleFieldMap3D"
    filename="HPS_b18d36_unfolded.dat"
     offsetX="1.0*cm"
     offsetY="2.0*cm"
     offsetZ="3.0*cm“
/>
```

Geant4 10.0 Improvements

- Would like to investigate the parallel geometry options in Geant4 to simulate more complex geometries
  - e.g. solenoid cryo chimney
- Do we need/want multi-threaded capabilities?
- Would use this opportunity to change default physics lists.
- What other new features from Geant4 10.0 would we like to benefit from?

Reconstruction

- Full reconstruction chain (e.g. DBD) involves a number of steps:
  - org.lcsim: track-finding and fitting
  - slicPandora: calorimeter clustering and track-cluster association, PFO creation
  - LCFI: vertex finding, flavor-tagging
- Improvements to the org.lcsim tracking ongoing, but at a low level, driven by need, limited by funding
- Maintaining interfaces to PandoraPFA and LCFI++ will require continuing effort.

Magnetic Field in org.lcsim

- A fourth-order Runge-Kutta stepper has been implemented in trf.
- Takes a track defined at a Surface and propagates to destination Surface
  - Input tracking tolerance drives adaptive step size
- Runge-Kutta propagation tested against helical propagation in constant field.
- Tested in piece-wise constant test fields where analytic solution can be calculated.

Track Fitting: Interactions

- trf package includes Interactor API to model the material effects encountered by charged particles traversing the detector.
  - Multiple Coulomb Scattering implemented as simple gaussian.
  - Bethe-Bloch with generic material for energy loss.
  - Both model material as generic X/X_0
  - OK for collider detector physics where most particles can safely be assumed to be pions.
- Introducing effects specific to individual particle IDs into package trfmat:
  - material-specific I, A, Z, δ for Bethe-Bloch
  - different terms for e^+ & e^-
  - Bremsstrahlung eloss
- Interactor can take track particle ID hypothesis as input.

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 (e.g. dE/dx and energy loss for electrons in particular)
- Small misalignments handled as perturbations on ideal surfaces
- Propagation to arbitrary planar surfaces in constant fields supported (not yet released).

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.
- Improvements will benefit SiD work when needed.

org.lcsim: Tracking

- Work ongoing within the context of HPS support to provide binding to GBL package.
- GBL provides tight coupling to millepede II alignment package.
- Modifications being made to compact.xml description of tracker elements to more naturally target survey and alignment parameters.
- Interested in learning from others' experience
  - e.g. EUDET telescope

org.lcsim : Calorimetry

- Modifications and improvements to support the SiD Ecal testbeam:
  - Use of hexagonal cells in both simulation and reconstruction
    - position to Cell ID
    - cell ID to position
    - cell ID provides List<CellID> neighbors
    - edge cells correctly handled
  - 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, but will eventually benefit use at other testbeams and LC.
- More robust conditions database handling

Housekeeping

- org.lcsim code repository moved from cvs to svn
- Broke up the monolithic lcsim package and disentangled some connections between lcsim and GeomConverter.
- LOT of code left behind
- Refactoring ongoing
- Leaner and cleaner code base
- Please contact us if needed functionality is missing or behavior has changed.

Computing

- Not currently engaged in any large-scale physics or detector simulations studies a la LOI, DBD, so computing needs are modest.
- Smaller-scale production manageable on individual clusters.
- Grid production available via ILCDirac
  - Dedicated GSE and GCE primarily provided by European institutions under LCG.
  - OSG support noticeably lacking.
- See Jan’s talk for plans for the future.

Future Plans

- Maintain existing simulation/reconstruction functionality.
- Continue to work towards increased software commonality.
  - But AIDA appears to be heading in a different direction with new Geant4-based simulation toolkit.
- Need to identify OSG support for large-scale production if we have future LOI/DBD-type studies.
- Need to find new institutional support for both software and computing effort.