News on Core Software Tools

Frank Gaede
DESY

ECFA Software & Physics Simulation
Phone Meeting, June 21, 2006

Outline

- new developments since Cambridge meeting
  - LCIO
  - Marlin/MarlinReco
  - Gear
  - Mokka

LCIO status and plans

- no major new features since Cambridge
  - minor bug fixes
  - command line tool in Java (Jeremy McCormick)
    - creation and comparison of random events
- under development:
  - vertex class in LCIO (see next slide)
  - transient object links (see below)
  - Track parameter definition (T.Kraemer)
    - writeup of L3 parameterization in LC-Note

Proposal for an LCIO vertex class

EVENT::Vertex

+ ~ Vertex()
+ *getMomentum() : const float**
+ *getMass() : float*
+ *getCharge() : float*
+ *getPosition() : const float**
+ *getCovMatrix() : const FloatVec&*
+ *getChi2() : float*
+ *getProbability() : float*
+ *getPreviousVertex() : Vertex**
+ *getParameters() : const FloatVec&*
+ *getTracks() : const TrackVec&*
+ *addTrack(track : Track*) : void*

EVENT::ReconstructedParticle

+ ~ ReconstructedParticle()
+ *getType() : int*
+ *isCompound() : bool*
+ *getMomentum() : const double**
+ *getEnergy() : double*
+ *getCovMatrix() : const FloatVec&*
+ *getMass() : double*
+ *getCharge() : float*
+ *getReferencePoint() : const float**
+ *getParticleIDs() : const ParticleIDVec&*
+ *getParticleIDUsed() : ParticleID**
+ *getGoodnessOfPID() : float*
+ *getParticles() : const ReconstructedParticleVec&*
+ *getClusters() : const ClusterVec&*
+ *getTracks() : const TrackVec&*
+ *addParticleID(pid : ParticleID*) : void*
+ *addParticle(particle : ReconstructedParticle*) : void*
+ *addCluster(cluster : Cluster*) : void*
+ *addTrack(track : Track*) : void*

- original LCIO idea:
  use ReconstructedParticle also for compound objects like jets and vertices
- LCFI proposes dedicated vertex class
- need to optimize to avoid overlap and
- redundancy with ReconstructedParticle

discussion: http://forum.linearcollider.org

LCIO transient object links

- long pending user request:
  - attach arbitrary objects to LCObjects
- needed during reconstruction:
  - fast and easy creation of links between various LCObject subtypes, eg. TrackerHits and Track
- aim: write code like:

```
struct MyPatRecLink : public LCObjectLinkTraits< TrackerHit> {} ;
...
Track* trk = ....
...
TrackerHit* h = dynamic_cast<TrackerHit>( col->getElementAt(i) ) ;
trk.link<MyPatRecLink>( h ) ;
...
TrackerHit* lh = trk.link<MyPatRecLink>() ;
```

Marlin Status and Plans

- no developments since Cambridge :(
- current version still: v00-09-04

- plans (still) :
  - improve dealing with and creation of steering files
    - show parameters in order specfied
    - provide minimal steering files
    - provide help for one single processor
  - user suggestions/ needs ?

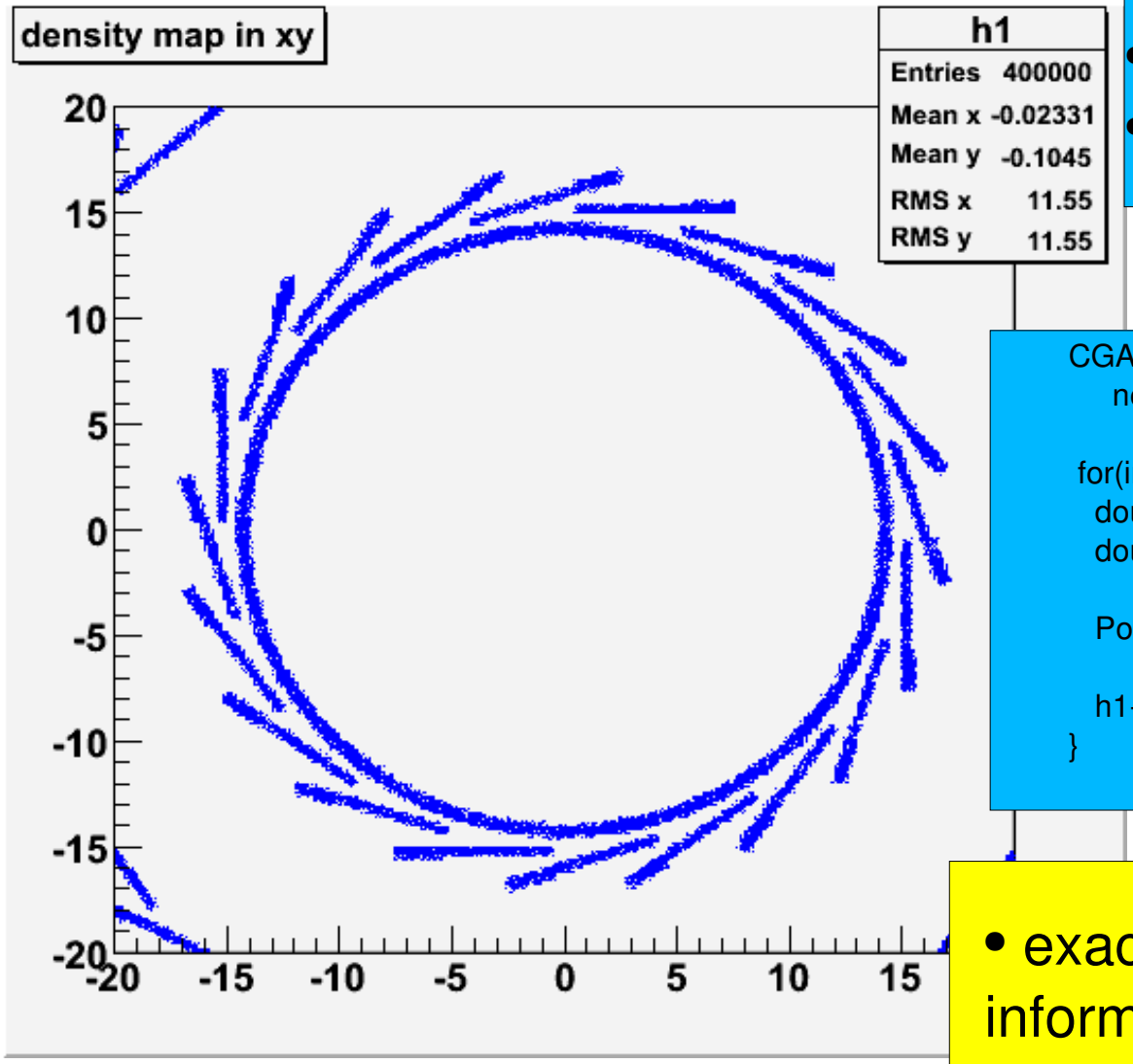
MarlinReco

- currently ongoing: restructuring of PFA code
  - improve code basis
    - utility code for geometry computations (helix to point,plane,...)
    - new abstract clustering classes/functions
  - improve physics quality of algorithms
    - need to demonstrate that PFA works
    - need to be better than calorimeter only @ 500 GeV

Gear status

- version v00-02 (Cambridge):
  - TPC, Hcal, Ecal interfaces defined and implemented
  - user parameters
- current 'release' v00-03-beta:
  - also write xml files from parameters in memory
  - tool to merge files: gearmerge
  - description of TPC prototypes (rectangular pad plane)
- planned release v00-03
  - first version of VTX description (R.Lippe)
  - input from Ringberg VTX workshop
  - first implemtation of GearCGA material properties (G.Mussat)

CGAGear



- implemented by G.Musat, LLR
- to be released soon

```
CGAGearPointProperties * pointProp =
    new CGAGearPointProperties(steer.str(),...);

 for(int i=0 ; i<nPoint ; ++i){
   double xr =  xmin +  ( xmax - xmin)  * random() ;
   double yr =  ymin +  ( ymax - ymin)  * random() ;

   Point3D p( xr, yr, z0 ) ;

   h1->fill(  xr, yr, pointProp->getDensity( p )  ) ;
 }
```

- exact geant4 material information at runtime !
- performance ?
- practical issues (linking g4) ?

GEAR – material properties

GearDistanceProperties

- GearDistanceProperties()
getMaterialNames(p0 : const Point3D&, p1 : const Point3D&) : const std::vector< std :: string >&
getMaterialThicknesses(p0 : const Point3D&, p1 : const Point3D&) : const std::vector< double >&
getNRadlen(p0 : const Point3D&, p1 : const Point3D&) : double
getNIntlen(p0 : const Point3D&, p1 : const Point3D&) : double
getBdL(pos : const Point3D&) : double
getEdL(pos : const Point3D&) : double

GearPointProperties

- GearPointProperties()
getCellID(pos : const Point3D&) : int
getMaterialName(pos : const Point3D&) : const std::string&
getDensity(pos : const Point3D&) : double
getTemperature(pos : const Point3D&) : double
getPressure(pos : const Point3D&) : double
getRadlen(pos : const Point3D&) : double
getIntlen(pos : const Point3D&) : double
getLocalPosition(pos : const Point3D&) : Point3D
getB(pos : const Point3D&) : double
getE(pos : const Point3D&) : double
getListOfLogicalVolumes(pos : const Point3D&) : std::vector< std :: string >
getListOfPhysicalVolumes(pos : const Point3D&) : std::vector< std :: string >
getRegion(pos : const Point3D&) : std::string
isTracker(pos : const Point3D&) : bool
isCalorimeter(pos : const Point3D&) : bool

- proposal since Argonne Simulation Meeting 2004
- implementation with Mokka CGA under development

Mokka

- new release 6.1 (G.Mussat, P. Mora-Freitas)

I. Monte Carlo truth information stored by defaut in the LCIO file.

II. HEPEvt ASCII event file format versus Monte Carlo truth information in LCIO.

III. New hepevt ASCII Brahms-like event files reader

IV. Possibility to choose a port number for the MySQL server host or a socket file for local connections

V. **Cooking the Model geometry at launch time**

VI. **First version of "MokkaGear" - automatic creation of gear files**

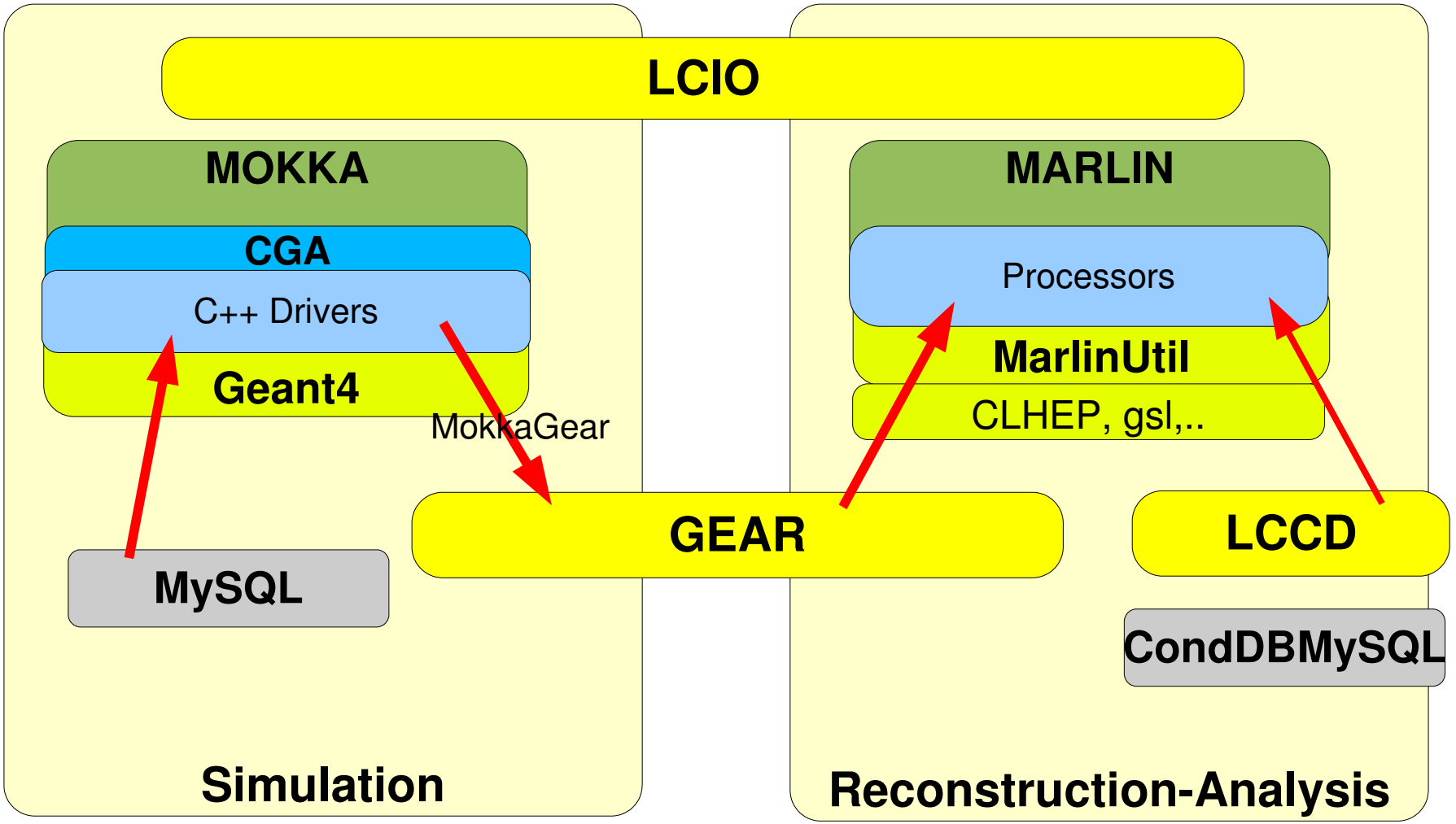
MokkaGear

- extension to Mokka (R.Lippe – Diploma Thesis)
- extract geometry information in drivers when detector is built
- use Gear to create XML files for reconstruction
- currently implemented:
  - TPC (tpc04), Ecal (ecal02) and Hcal (hcal04)
- **released with Mokka 6.1**
- optional feature
  - only if Gear is installed and included

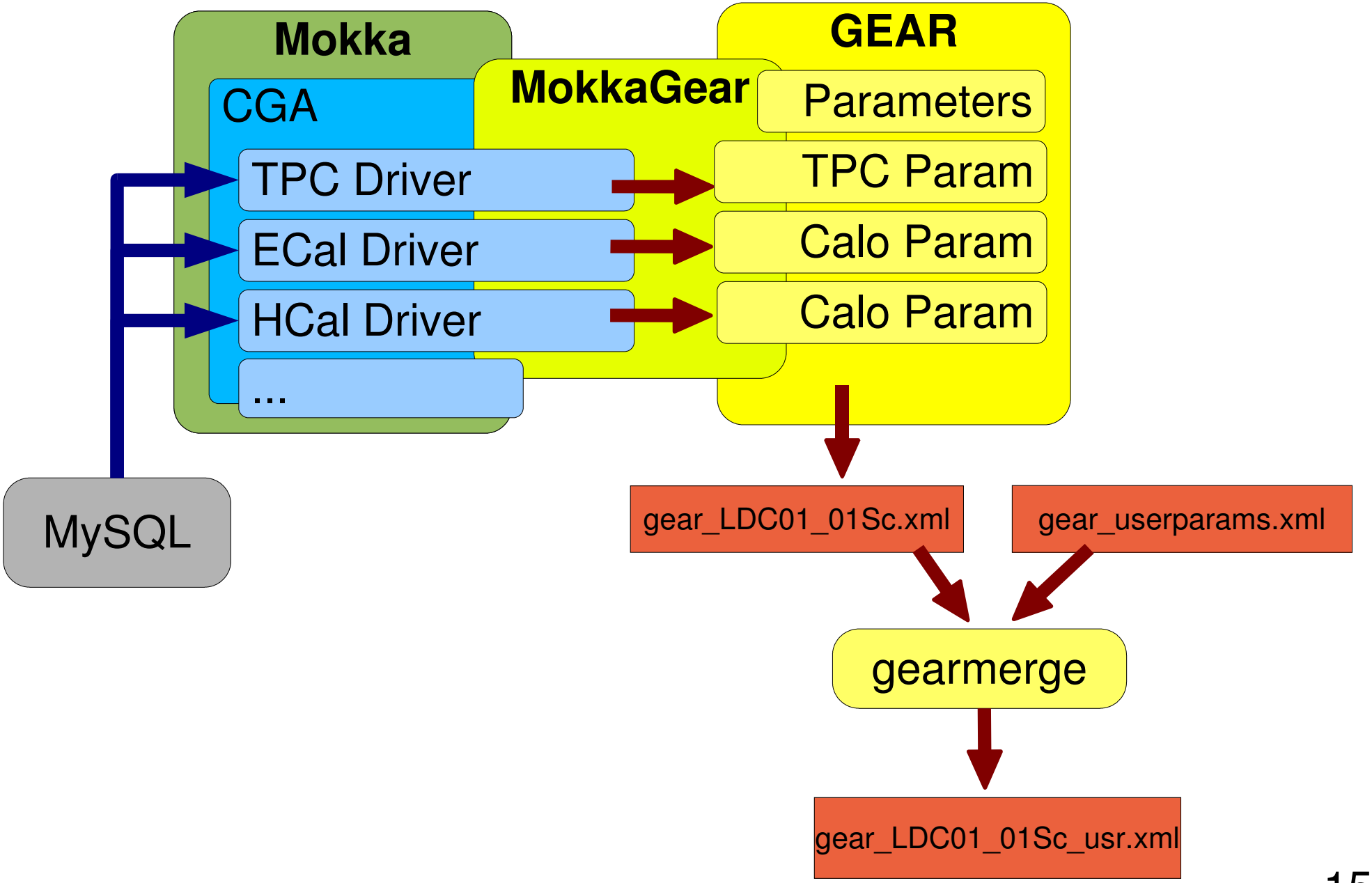
aim: have only one source of of information
for describing the detector geometry !

Backup slides ...

LDC simulation framework



MokkaGear


Mokka
CGA
TPC Driver
ECal Driver
HCal Driver
...
MokkaGear
GEAR
Parameters
TPC Param
Calo Param
Calo Param
MySQL
gear_LDC01_01Sc.xml
gear_userparams.xml
gearmerge
gear_LDC01_01Sc_usr.xml