LCIO
The data model of the persistency framework for LC detector simulation

Frank Gaede, DESY, IT
4th ECFA/DESY LC Workshop
Amsterdam April 1st-4th 2003

Outline

- Introduction
- Software/Status (brief, see talk in simulation session)
- Data model
  - simulation
  - reconstruction
- Summary
**Introduction**

- At Prague workshop decided to have
  **Data format/persistency task force:**
  "Define an abstract object persistency layer and a data model for linear collider simulation studies until the Amsterdam workshop."

- People:
  - Ties Behnke  - DESY/SLAC
  - Frank Gaede  - DESY
  - Norman Graf  - SLAC
  - Tony Johnson  - SLAC
  - Paulo Mora de Freitas  - IN2P3

---

**Motivation**

**LCIO Persistency Framework**

- Generator
- Simulation
- Reconstruction
- Analysis

**geometry**

LCIO, 4th ECFA/DESY Workshop, Amsterdam 2003

Frank Gaede, DESY IT
The Persistency Framework

Meetings

- Meeting at SLAC 12/09-12/13/02:
  (T. Behnke, F. Gaede, N. Graf, T. Johnson)
  - agreement to have common persistency framework in one US group (hep.lcd) and in the European group: LCIO
  - agreement on the (first) implementation format
  - first definition of the data model

- Meeting at Ecole Polytechnique 01/14-01/15/03:
  (F. Gaede, P. Mora de Freitas, H. Videau, J.-C. Brient)
  - agreement to use LCIO as the output format for the Mokka simulation framework
  - further discussions and refinement of the data model (reconstruction)

- Presentation and discussion of the data model at CERN miniworkshop of detector performance group 25/02/03
- Several phone meetings
Status of LCIO

- need Java, C++ and f77 interface
- C++ implementation ready for testing (sim.)
  - currently implemented into Mokka framework
- f77 prototype
- Java development underway
- SIO as persistency format
- data model for simulation and reconstruction
  - simulation (settled)
  - reconstruction (initial proposal)

Overview of the Data Model:

- RunHeader
- SimHeader
- RecoHeader
- Event
- MCParticle
- TrackerHit
- CalorimeterHit
- Reco
- Track
- Cluster
- ReconstructedObject
- ReconstructedParticle
- SIM
Data model - LCRunHeader

- **block**: RunHeader
  - int: runNumber
  - string: detectorName
  - string: description
  - string[]): activeSubdetectors

=> describes the run setup

Data model - LCEventHeader

- **EventHeader**
  - int: runNumber
  - int: evtNumber
  - string: detectorName
  - String[]): subdetectorName
  - blockNames:
    - string: blockName
    - string: blockType

=> describes event data –  
needed for fast skip
Data model – LCEvent (sim)

- **Event**
  - int: runNumber
  - int: evtNumber
  - string: detectorName
  - String[]: subdetectorName
  - long: timeStamps

- **MCParticle**
  - ptr: parent
  - ptr: secondParent
  - ptr[]: daughters
  - int: pdgid:
  - int: hepevtStatus
    - (0, 1, 2, 3 HepEvt)
    - (201, 202 sim. decay)

- **MCParticle cont.**
  - double[3]: start
    - (production vertex)
  - float[3]: momentum
    - (at vertex)
  - float: energy
  - float: charge
Data model - LCEvent (sim)

- TrackerHit
- string: subdetector
  - int: hitFlags (detector specific: Id, key, etc.)
  - double[3]: position
  - float: dEdx
  - float: time
  - ptr: MCParticle

Data model - LCEvent (sim)

- CalorimeterHit
- string: subdetector
  - int: cellId0
  - int: cellId1
  - float: energy
  - float[3]: position – optional (file size)
  - particle contributions:
    - ptr: MCParticle
    - float: energyContribution
    - float: time
    - int: PDG (of secondary) - optional
Data model - LCEvent (reco)

- **OutputHeader**
  - int: isrFlag
  - float: colliderEnergy
  - int: flag0 (to be defined)
  - int: flag1 (to be defined)
  - int: reconstructionProgramTag
  - float: Bfield

  --> could be combined with global header…

Data model - LCEvent (reco)

- **Track**
  - int: tracktype (full reconstr, TPC only, Muon only, etc.)
  - float: momentum \( \frac{1}{p} \)
  - float: theta
  - float: phi
  - float: charge
  - float: d0 (Impact Parameter in r-phi)
  - float: z0 (Impact Parameter in r-z)
  - float[15]: cov.matrix
  - float: reference point \((x, y, z)\)
  - float: chi**2 of fit
  - float[10]: dEdx (weights and probabilities)
  - TrackerHits: - optional
    - ptr: TrackerHit
Data model - LCEvent (reco)

- **Cluster**
  - int: detector (type of cluster: ECAL, HCAL, combined...)
  - int: clustertype (neutral, charged, undefined cluster)
  - float: energy
  - float[3]: position (center of cluster x, y, z)
  - float[6]: errpos (cov. matrix of position)
  - float: theta (intrinsic direction: theta at position)
  - float: phi (intrinsic direction: phi at position)
  - float[3]: errdir (cov. matrix of direction)
  - float[6]: shapeParameters (definition needed)
  - float[3]: weights (compatible with cm., had., muon)
  - CalorimeterHits: - optional
    - ptr: CalorimeterHit
    - float: contribution

- **ReconstructedParticle**
  - int: primaryFlag (0: secondary, 1: primary)
  - int: ObjectType (charged/neutral particle)
  - float[3]: 3-Vec (px, py, pz)
  - float: energy
  - float[10]: covariancematrix
  - float: charge
  - float[3]: reference position for 4-vector
  - float[5]: PID_type (hypotheses for e, g, pi, K, p, ...)

- **ReconstructedParticle cont.**
  - Tracks:
    - ptr: Track
    - float: weight
  - Clusters:
    - ptr: Cluster
    - float: weight
  - MCParticles:
    - ptr: MCParticle
    - float: weight

Have separate MC-link object?
Data model - LCEvent (reco)

- **ReconstructedObject**
  - int: ObjectType (jet, vertex, ...)
  - float[5]: 4vec (4-vector of object (px, py, pz, E, M))
  - float[3]: reference (position)
  - float[15]: covariance matrix
  - reconstructedParticle:
    - ptr: reconstructedParticle
    - float: weight

=> generic reconstructed objects, linked to reconstructed particles

Summary

- LCIO is a persistency framework for linear collider simulation software
  - Java, C++ and f77 user interface
  - currently implemented in simulation frameworks:
    - hep.lc
    - Mokka/BRAHMS-reco

- datamodel for simulation and reconstruction output
  -> comments welcome!

- see LCIO homepage for more details: