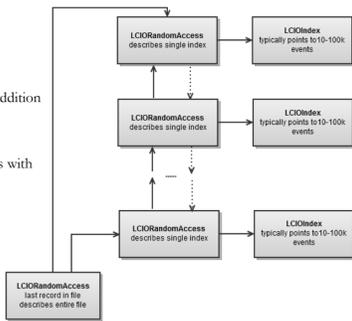


LCIO provides an experiment-independent hierarchical **event data model** and a **persistence solution (I/O)** for HEP. Developed collaboratively between SLAC & DESY beginning in 2002, it is being used by the international Lepton Collider community both for physics and detector response simulations and also in numerous detector R&D testbeam experiments. Standardizing on a simple, but performant, EDM and I/O has allowed event sharing and code reuse between regions, operating systems, and languages.



LCIO Random Access Support

- ❖ Allow efficient access to specific events in LCIO files. Events should be selectable by
 - Run
 - Run + Event
 - Index (i.e. 10000th event in file)
 - Tag (e.g. EMISS>200)
- ❖ Access must work for "chains" of files in addition to individual files.
- ❖ Must scale to support very large event sets
 - Complete index may not fit in memory
- ❖ Must still be possible to read and write files with only sequential access
- ❖ Adds 2 new record types
 - Backwards compatible
 - Can easily be added to existing files
 - Forward pointers are optional
 - Files can be written sequentially
- ❖ Support added in Java and C++ implementations



1D and 2D TrackerHits

- ❖ **TrackerHitPlanar**
 - x, y, z - 'space point'
 - u(theta, phi), v(theta, phi) – measurement directions (spanning vectors in the plane)
 - du, dv - measurement errors
 - to be used for 1D and 2D
- ❖ **TrackerHitCylindrical**
 - x, y, z - 'space point'
 - R, Xc, Yc – cylinder parameters (parallel to z)
 - dphi, dz - measurement errors
 - to be used for 1D and 2D
- ❖ these also implement the TrackerHit interface (x,y,z, cov) for backward compatibility and code reusability

Track & TrackStates

- ❖ Track now accommodates multiple track states, representing fits to the trajectory at various locations.
 - E.g. @IP, @first/last hit, @vertex, @calorimeter.
 - ❖ TrackState returned either by identifier or closest to a given point.
- ```

virtual ~TrackState ()
Destructor.
virtual int getLocation () const =0
The location of the track state.
virtual float getD0 () const =0
Impact parameter of the track in (r-phi).
virtual float getPhi () const =0
Phi of the track at the reference point.
virtual float getOmega () const =0
Omega is the signed curvature of the track in (1/mm).
virtual float getZ0 () const =0
Impact parameter of the track in (r-z).
virtual float getTanLambda () const =0
Lambda is the dip angle of the track in r-z at the reference point.
virtual const FloatVec & getCovMatrix () const =0
Covariance matrix of the track parameters.
virtual const float * getReferencePoint () const =0
Reference point of the track parameters.
virtual const TrackStateVec & getTrackStates () const =0
Returns track states associated to this track.
virtual const TrackState * getClosestTrackState (float x, float y, float z) const =0
Returns track state closest to the given point.
virtual const TrackState * getTrackState (int location) const =0
Returns track state for the given location - or NULL if not found.
virtual const TrackerHitVec & getTrackerHits () const =0
Optionally (check/set flag(LCIO:TRBIT_HITS)=1) return the hits that have been used to create this track.

```

### Miscellaneous Improvements

- ❖ Extensions to **MCParticle**
  - added spin and color flow information
  - Whizard event generator will write out MC Event in LCIO format.
- ❖ **SimCalorimeterHit**
  - optionally add the position where the energy deposition (step) occurred:
    - float[3] getStepPosition( int i ) enabled only if flag bit is set
    - useful for detailed simulation studies of edge effects in calorimeter
- ❖ **TrackerHit and CalorimeterHit**
  - canonical way of accessing layer number
    - getLayerNumber(), setLayerNumber()
  - Add ID to all hits to allow access to geometry system
- ❖ **Cluster**
  - Add energy uncertainty and time

### LCIO & ROOT

- ❖ LCIO v02-00 allows to optionally create a ROOT dictionary for all LCIO classes. With this one can:
  - use LCIO classes in ROOT macros
  - write simple ROOT trees, e.g. std::vector<MCParticleImp\*>
  - use TTreeDraw for quick interactive analysis of LCOjects, e.g.:
 

```

//---gamma conversions:
TCut isPhoton("MCParticlesSkimmed.getPDG()==22");
LCIO->Draw("MCParticlesSkimmed._endpoint[[0]:
MCParticlesSkimmed._endpoint[[1]",isPhoton) ;

```
  - write complete LCIO events in one ROOT branch
- ❖ see: \$LCIO/examples/cpp/rootDict/README for details & help
- ❖ We are interested in feedback from end users!

### Code Infrastructure

- ❖ LCIO repository moved to SVN
  - Web interface:
    - http://java.freehep.org/svn/repos/lcio/list/?revision=HEAD
  - Checkout:
    - svn co svn://svn.freehep.org/lcio/tags/v02-00 v02-00
- ❖ C++ build now uses **cmake**
- ❖ Java build uses **maven**
- ❖ Documentation available at
  - http://lcio.desy.de/