

User Conventions for LCIO

LC software workshop
Hamburg, 27/28 June 2005

Jürgen Samson

Outline

- freedoms/arbitrariness when using LCIO
- need of conventions
- what to do

LCIO vs. Modularity

- LCIO
 - designed to fit the needs of LC development
 - general enough to cover any detector design
- Marlin
 - provides modularity
- users wish
 - write software once, apply to many (different) LCIO files
 - switch from MC to real data
 - concept studies:
change detector component; change only reco. modules directly affected by this change

The Problem

- ambiguities in LCIO
 - meaning of certain indices is left to the user
 - flexible design of LCIO does not fix the structure of an event
- the same problem with MARLIN
 - many ways to pass information from one processor to another
 - names of collections
 - “granularity” of information stored in LCIO file

Meaning of Indices, Flagwords

LCIO: EVENT::Cluster class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lcio.desy.de/v01-05/doc/doxygen_api/html/classEVENT_1_1Cluster.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [Related Pages](#)

EVENT::Cluster Class Reference

The LCIO cluster. [More...](#)

```
#include <Cluster.h>
```

Inheritance diagram for EVENT::Cluster:

```
graph BT; IOIMPL::ClusterIOImpl --> IMPL::ClusterImpl; IMPL::ClusterImpl --> EVENT::Cluster; EVENT::Cluster --> EVENT::LCObject;
```

[List of all members.](#)

Public Member Functions

virtual `~Cluster ()`
Destructor.

virtual int `getType () const=0`
Flagword that defines the type of cluster.

virtual const ClusterVec & `getClusters () const=0`
The clusters that have been combined to this cluster.

virtual const CalorimeterHitVec & `getCalorimeterHits () const=0`
The hits that have been combined to this cluster.

virtual const FloatVec & `getHitContributions () const=0`
Returns the energy contribution of the hits Runs parallel to the CalorimeterHitVec from `getCalorimeterHits()`.

virtual const FloatVec & `getSubdetectorEnergies () const=0`
A vector that holds the energy observed in a particular subdetectors.

Done Adblock

LCIO Parameters

LCIO: IMPL::LCEventImpl class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lcio.desy.de/v01-04/doc/doxygen_api/html/classIMPL_1_1LCEventImpl.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [Related Pages](#)

IMPL::LCEventImpl Class Reference

Implementation of the main event class. [More...](#)

```
#include <LCEventImpl.h>
```

Inheritance diagram for IMPL::LCEventImpl:

```
graph BT; IOIMPL::LCEventIOImpl --> IMPL::LCEventImpl; IMPL::LCEventImpl --> EVENT::LCEvent; IMPL::LCEventImpl --> IMPL::AccessChecked;
```

[List of all members.](#)

Public Member Functions

virtual	<code>~LCEventImpl ()</code>	Destructor.
virtual int	<code>getRunNumber () const</code>	Return the run number off this event.
virtual int	<code>getEventNumber () const</code>	Returns this event's number .
virtual	<code>removeCollection (const EVENT::ReadOnlyException, std::exception)</code>	Removes (and deletes) the collection with name (if it exists in the event).
virtual const EVENT::LCParameters &	<code>getParameters () const</code>	Parameters defined for this event.
virtual EVENT::LCParameters &	<code>parameters ()</code>	Parameters defined for this run.

Done Adblock

LCIO Parameters

LCIO: EVENT::LCParameters class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lcio.desy.de/v01-04/doc/doxygen_api/html/classEVENT_1_1LCParameters.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [Related Pages](#)

EVENT::LCParameters Class Reference

Public Member Functions

virtual [~LCParameters](#) ()
Destructor.

virtual int [getIntVal](#) (const std::string &key) const=0
Returns the first integer value for the given key.

virtual float [getFloatVal](#) (const std::string &key) const=0
Returns the first float value for the given key.

virtual const std::string & [getStringVal](#) (const std::string &key) const=0
Returns the first string value for the given key.

virtual IntVec & [getIntVals](#) (const std::string &key, IntVec &values) const=0
Adds all integer values for the given key to values.

virtual void [setValue](#) (const std::string &key, int value)=0
Set integer value for the given key.

virtual void [setValue](#) (const std::string &key, float value)=0
Set float value for the given key.

virtual void [setValue](#) (const std::string &key, const std::string &value)=0
Set string value for the given key.

virtual void [setValues](#) (const std::string &key, IntVec &values)=0
Set integer values for the given key.

virtual void [setValues](#) (const std::string &key, FloatVec &values)=0
Set float values for the given key.

virtual void [setValues](#) (const std::string &key, StringVec &values)=0
Set string values for the given key.

Detailed Description

Done Adblock

How to Handle Indices, etc.

- don't hard code the numbers
 - use at least constants in a header file
 - better: extract the numbers at runtime from the parameter sections
- ⇒ Naming conventions, not numbering conventions

Collection: Cluster type:Cluster size:50 flags:0
ClustertypeBits: ECALBIT:1, HCALBIT:2, COMBBIT:3

Type	Energy	Position	ITheta	IPhi
8	3.8000	[0.0000,0....	.26952	2.4951
8	3.8000	[0.0000,0....	2.3066	2.5265
8	1.8334	[0.0000,0....	3.0929	1.2093
8	4.9168	[0.0000,0....	1.1732	4.4074
8	7.9544	[0.0000,0....	1.6472	5.0725
8	1.1312	[0.0000,0....	.19369	2.0014
8	4.2729	[0.0000,0....	.74988	5.2875

Analyzed 1 records in 268ms 4.23/6.12MB

Keywords

- keywords have arbitrary names
 - naming conventions are needed
 - some keywords relevant for physics
 - > ask LCIO authors to provide string constants

The screenshot shows the JAS3 software interface. The main window displays event data for Run:0 Event:3. The left pane shows a tree view of data collections, with 'RecoParticles' selected. The right pane shows the details of the selected collection, including its type, size, and flags, followed by a table of particle properties.

Collection: RecoParticles type:ReconstructedParticle size:42 flags:0
sphericity: 0.2904724
aplanarity: 0.007926704
sphericity_tensor_eigenvalues: 0.0052844696, 0.18836379, 0.8063519

Type	Momentum	Energy	Mass	Charge	...
2	[-.70783, .15518, .13824]	.75088	.14000	-1.0000	...
2	[-.57379, .38106, .24451]	.74419	.14000	-1.0000	...
2	[-.51748, .30814, .26902]	.67432	.14000	1.0000	...
2	[-.16672, -.56989, -.33544]	.69619	.14000	1.0000	...
2	[-.53791, .18521, .41179]	.71611	.14000	-1.0000	...
2	[-.24100, -.44827, -.13781]	.54555	.14000	-1.0000	...
2	[-.26666, .36965, .031661]	.47786	.14000	-1.0000	...
2	[-.28908, .33022, .58256]	.74269	.14000	-1.0000	...
2	[-.062331, -.41699, -.64149]	.78030	.14000	-1.0000	...
2	[.14222, .21663, .084962]	.30655	.14000	-1.0000	...
1	[-1.1020, -.0013680, 309281]	1.1446	0.0000	1.0000	...

Analized 1 records in 5405ms

Structure of Events

- Task:
“Get the energy, which a reconstructed particle deposited in the e-cal”
- Problem:
 - LCIO does not want to force an detector to have an e-cal and h-cal (?)
 - thus LCIO cannot fix the way to store this information

ReconstructedParticle

LCIO: EVENT::ReconstructedParticle-class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lcio.desy.de/v01-05/doc/doxygen_api/html/classEVENT_1_1ReconstructedParticle.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [Related Pages](#)

EVENT::ReconstructedParticle Class Reference

The LCIO reconstructedParticle. [More...](#)

```
#include <ReconstructedParticle.h>
```

Inheritance diagram for EVENT::ReconstructedParticle:

```
graph BT; IOIMPL[IOIMPL::ReconstructedParticleIOImpl] --> IMPL[IMPL::ReconstructedParticleImpl]; IMPL --> EVENT[EVENT::ReconstructedParticle]; EVENT --> LCOBJECT[EVENT::LCObject];
```

[List of all members.](#)

Public Member Functions

virtual	~ReconstructedParticle ()	Destructor.
virtual int	getType () const=0	Type of reconstructed particle.
virtual bool	isCompound () const=0	Convenient method - same as (getParticles() size() > 0)
virtual const ReconstructedParticleVec &	getParticles () const=0	The reconstructed particles that have been combined to this particle.
virtual const ClusterVec &	getClusters () const=0	The clusters that have been used for this particle.
virtual const TrackVec &	getTracks () const=0	The tracks that have been used for this particle.
virtual void	addParticleID (ParticleID *pid)=0	

Done Adblock

Cluster

LCIO: EVENT::Cluster class Reference - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://lcio.desy.de/v01-05/doc/doxygen_api/html/classEVENT_1_1Cluster.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [Related Pages](#)

EVENT::Cluster Class Reference

The LCIO cluster. [More...](#)

```
#include <Cluster.h>
```

Inheritance diagram for EVENT::Cluster:

```
graph BT; IOIMPL::ClusterIOImpl --> IMPL::ClusterImpl; IMPL::ClusterImpl --> EVENT::Cluster; EVENT::Cluster --> EVENT::LCObject;
```

[List of all members.](#)

Public Member Functions

virtual `~Cluster ()`
Destructor.

virtual int `getType () const=0`
Flagword that defines the type of cluster.

virtual const ClusterVec & `getClusters () const=0`
The clusters that have been combined to this cluster.

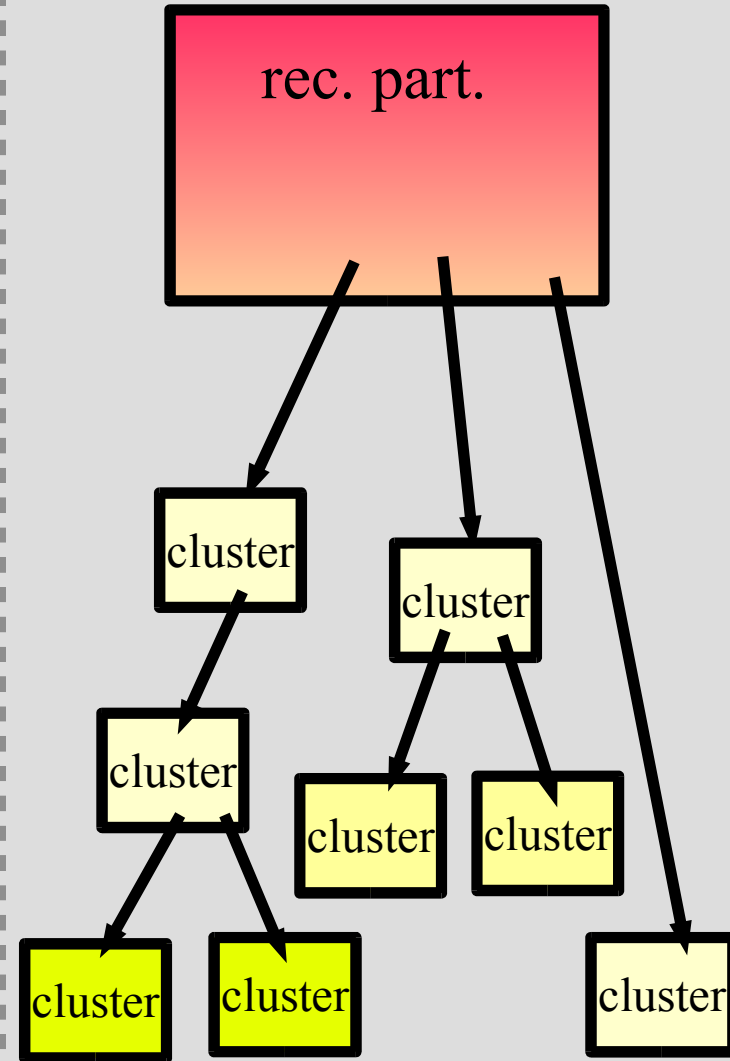
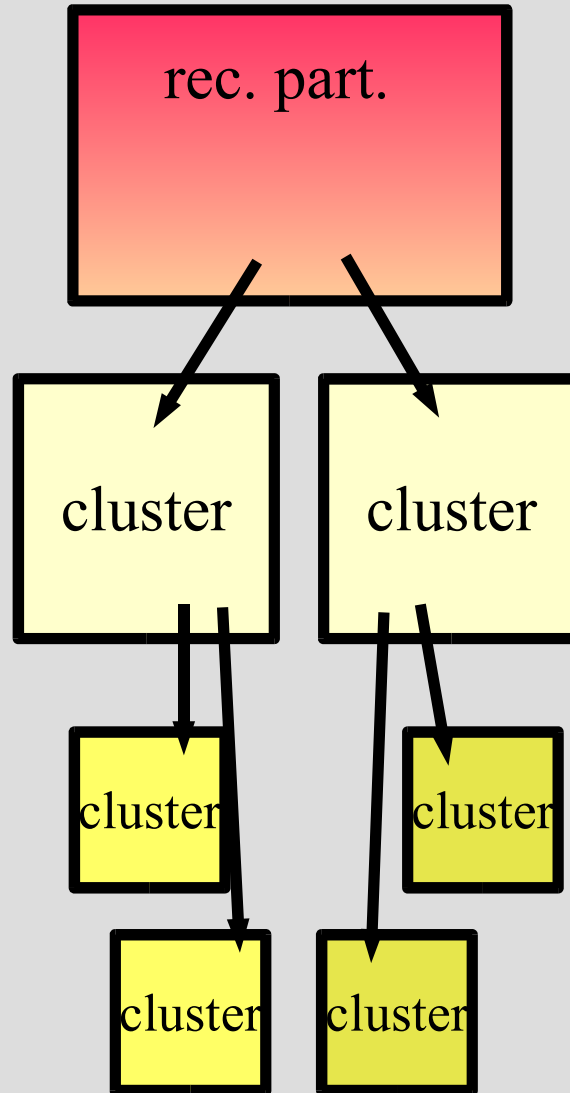
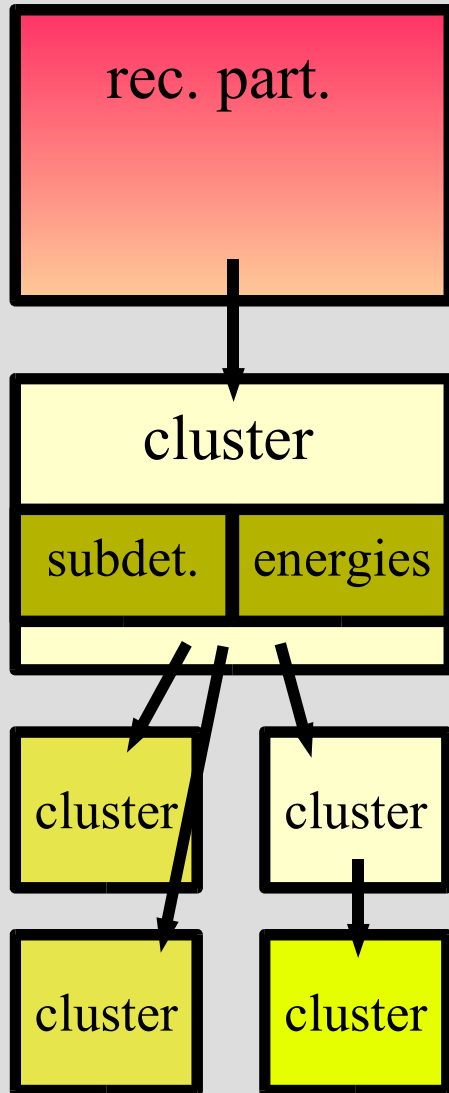
virtual const CalorimeterHitVec & `getCalorimeterHits () const=0`
The hits that have been combined to this cluster.

virtual const FloatVec & `getHitContributions () const=0`
Returns the energy contribution of the hits Runs parallel to the CalorimeterHitVec from `getCalorimeterHits()`.

virtual const FloatVec & `getSubdetectorEnergies () const=0`
A vector that holds the energy observed in a particular subdetectors.

Done Adblock

Several Possibilities



Modularity

- MARLIN modules have to process LCIO events prepared by other modules/software
 - collection names must be known
 - data needed by a processor must be in the event
 - event structure must be compatible
 - processors should be able to change “boundary conditions” (tpc radius, calorimeter cell sizes, ...)

Conclusions

- e.g. MARLIN processors:
 - provide steering parameter to change collection names
 - provide steering parameters for external parameters (or use geometry API) if possible
- but:
 - providing steering parameters for everything doesn't seem to be feasible
 - changing default values (e.g. collection names) for every processor can be very annoying
 - some input for a processor (needed collections, etc.) cannot be changed

Conclusions

=> conventions are needed

- useful conventions emerge from actual implementations
=> there are (many) candidates for conventions

Todo (i)

- find conventions
 - talk early to other software developers (not just those in the office next to you)
 - figure out where conventions are needed
 - find out which conventions seem to be useful / practicable
 - single out one (or a few) convention(s)
(a small set of conventions can be hidden by helper functions “getEcalEnergy()”)

Todo (ii)

- documentation
 - create “self documenting” LCIO files
i.e. put meaning of bitfields, indices, etc. in the parameter block of collections, events, runs
 - write human readable conventions
i.e. write down which input your processors exactly needs
 - gather several conventions at a central place
(provide helper functions, libraries, ...)
 - write software to check whether a LCIO file follows certain conventions