Overview of RawCalorimeterHit and its modification

Taikan Suehara
(Kyushu University)

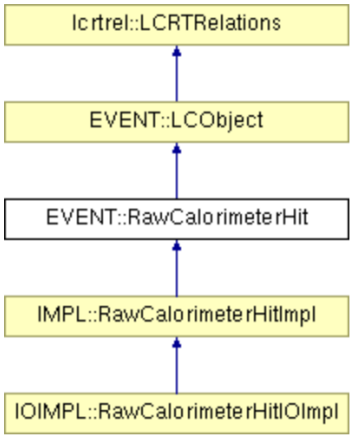
RawCalorimeterHit

EVENT::RawCalorimeterHit Class Reference

The generic calorimeter hit for real data (or simulation thereof). More...

#include <pre-generated/EVENT/RawCalorimeterHit.h>

Inheritance diagram for EVENT::RawCalorimeterHit:



List of all members.

Public Types

typedef RawCalorimeterHit	**lcobject_type** Useful typedef for template programming with **LCIO**.

Public Member Functions

virtual	**~RawCalorimeterHit** () Destructor.
virtual int	**getCellID0** () const =0 Returns the detector specific (geometrical) cell id.
virtual int	**getCellID1** () const =0 Returns the second detector specific (geometrical) cell id.
virtual int	**getAmplitude** () const =0 Returns the amplitude of the hit in ADC counts.
virtual int	**getTimeStamp** () const =0 Returns a time stamp for the hit.

Simple object having 4 integers

- CellID0
- CellID1
- Amplitude
- TimeStamp

Not enough?

CalorimeterHit

Public Member Functions

virtual	**~CalorimeterHit** () Destructor.
virtual int	**getCellID0** () const =0 Returns the detector specific (geometrical) cell id.
virtual int	**getCellID1** () const =0 Returns the second detector specific (geometrical) cell id.
virtual float	**getEnergy** () const =0 Returns the energy of the hit in [GeV].
virtual float	**getEnergyError** () const =0 Returns the error of the hit energy in [GeV].
virtual float	**getTime** () const =0 Returns the time of the hit in [ns].
virtual const float *	**getPosition** () const =0 Returns the position of the hit in world coordinates.
virtual int	**getType** () const =0 Type of hit.
virtual **LCObject** *	**getRawHit** () const =0 The **RawCalorimeterHit**.

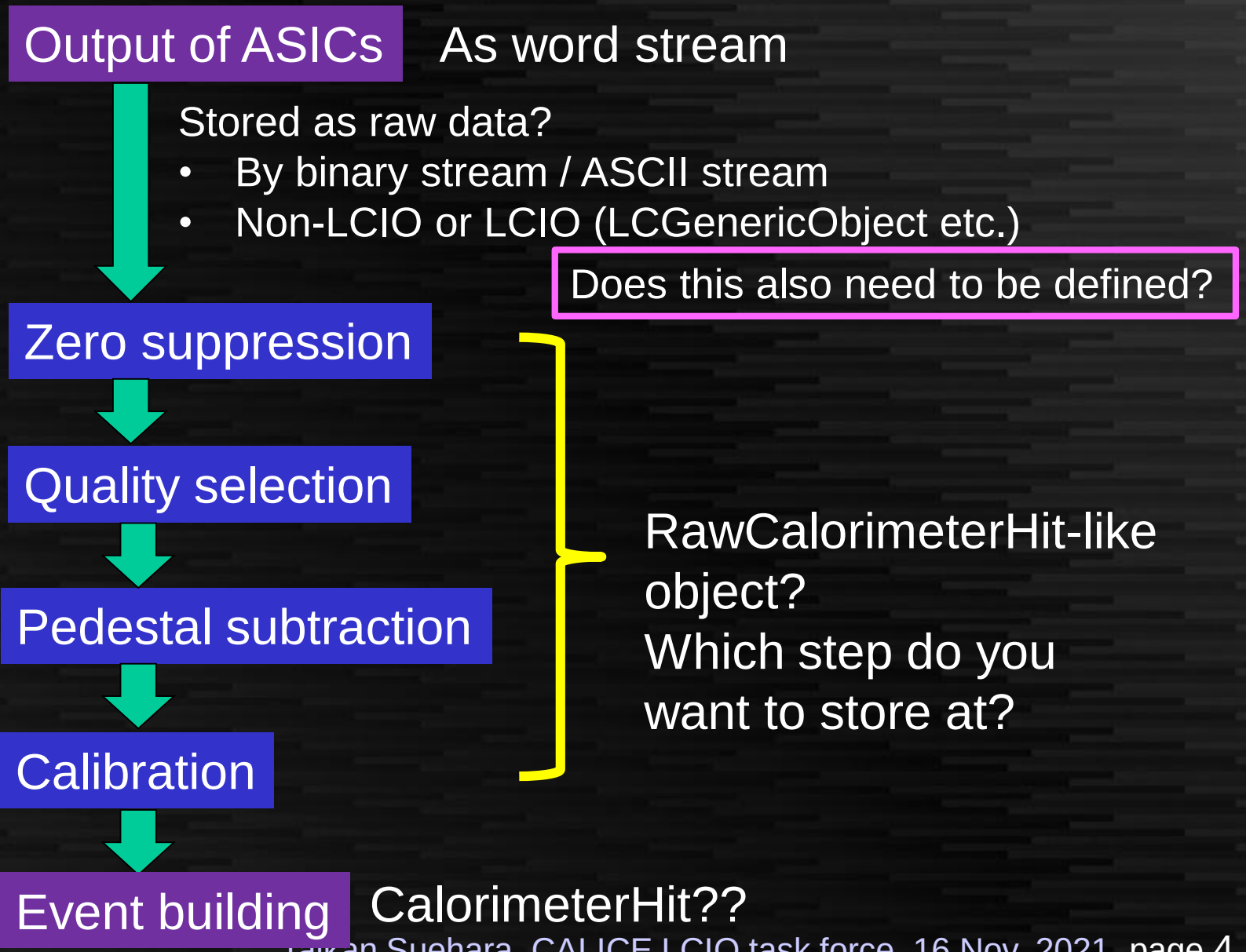
Integers
- CellID0
- CellID1
- Type

Floats
- Energy
- EnergyError
- Time
- Position (x,y,z)

Pointer
- RawCaloHit

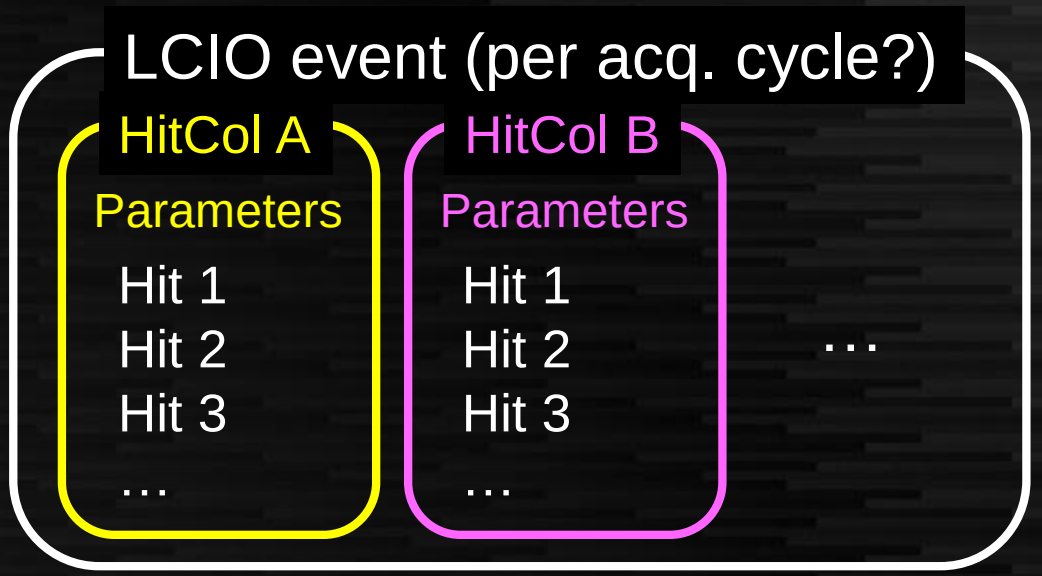
Used for eg. ILD simulation → not easy to modify

Event data flow (in my understanding)



Discussion points

- What do you need in addition
  - To the data structure
  - To the parameters (flexible: no need of common def.)
- Class name (for backward compatibility)
  - Should we change the class name?
  - To what?
- Others?



Possible schedule

- Discussion 1 (today)
  - Collecting info from each subsystem
- Discussion 2 (Dec.)
  - Discussion of first proposal
- Discussion 3 (Jan.)
  - Proposal to be fixed
    → Discussion with LCIO experts
- Discussion 4 (after the discussion with experts)
  - Final decision?
  - Starting to write a summary document