Task force report: RawCalorimeterHit modification

Taikan Suehara
(Kyushu University)

Motivation of the new structure

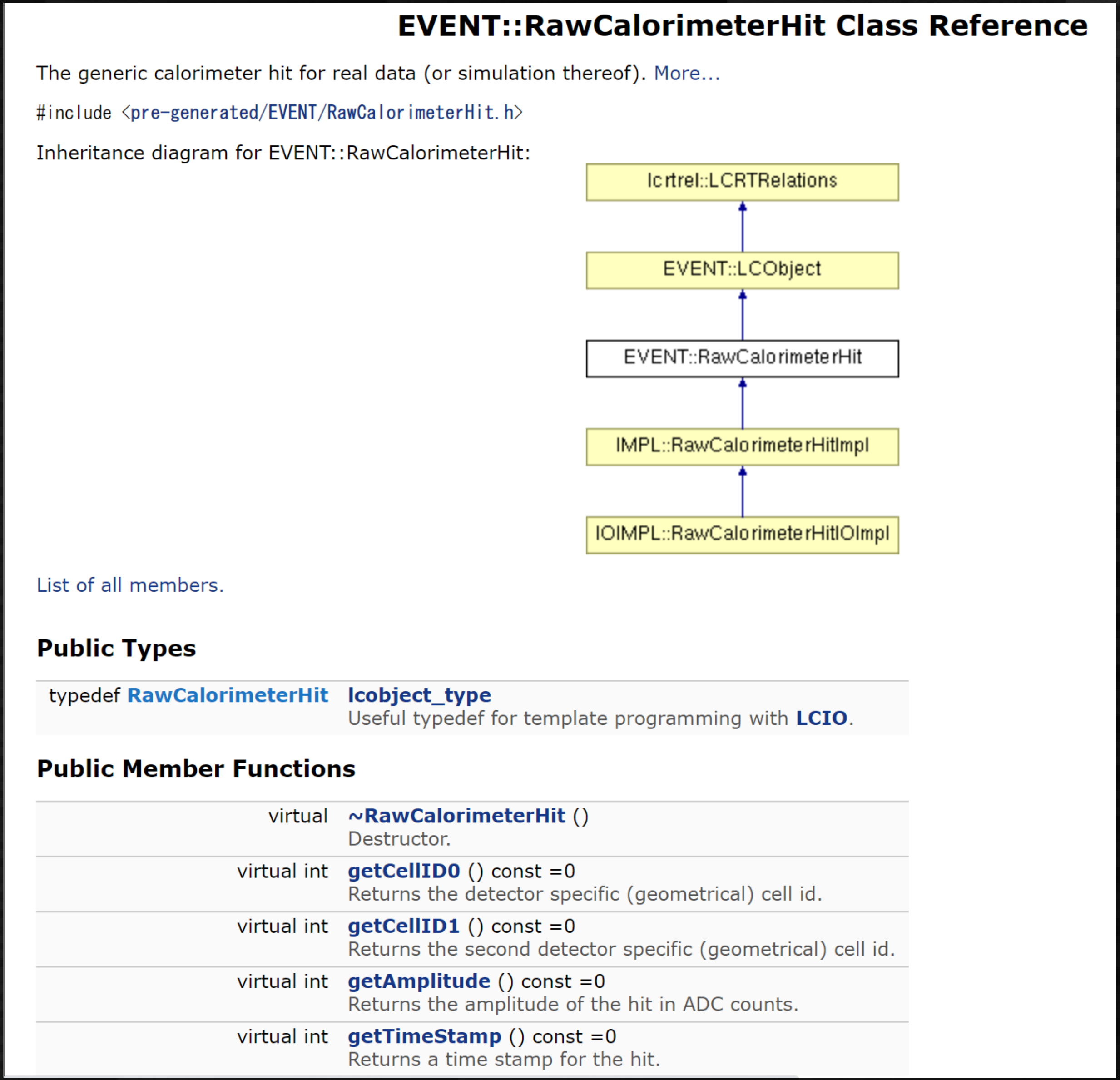
- Storage of test beam data
  - Array-like object (eg. ROC output)
  - Hit-like object
    - Before event building – RawCalorimeterHit
    - After event building, calibration etc. – CalorimeterHit
- RawCalorimeterHit is not enough for holding all data with convenient ways
  → Structure should be modified to accommodate necessary data from all subsystems

Task force members and meetings

- Chair (DAQ): T. Suehara
- SiW-ECAL: A. Irles
- SciCAL: J. Kvasnicka, K. Braggi, W. Ootani
- SDHCAL: G. Grenier
- Software: D. Jeans

- 3 meetings so far
  - Sep. 29, 2021: kick-off
  - Nov. 16, 2021: collect information from subsystems
  - Feb. 23, 2022: proposal and discussion with software experts (F. Gaede, T. Madlener)

Current 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

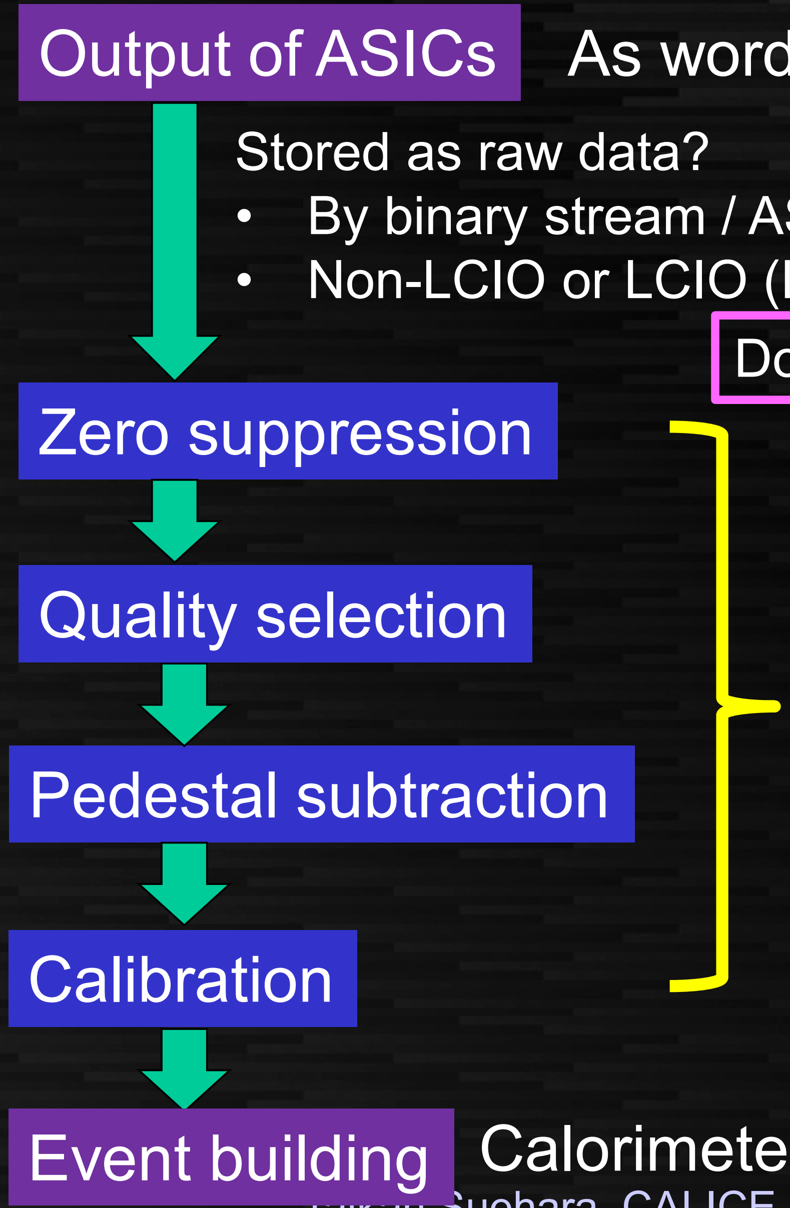
virtual	**~RawCalorimeterHit** () Destructor.
virtual int	**getCellID0** () const =0 Returns the detector specific (geometrical) cell id.
virtual int	**getCellID1** () const =0 Returns the second detector specific (geometrical) cell id.
virtual int	**getAmplitude** () const =0 Returns the amplitude of 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?

Event data flow (in my understanding)



Output of ASICs

As word stream

Stored as raw data?

- By binary stream / ASCII stream
- Non-LCIO or LCIO (LCGenericObject etc.)

Does this also need to be defined?

Zero suppression

Quality selection

Pedestal subtraction

Calibration

RawCalorimeterHit-like object?
Which step do you want to store at?

Event building

CalorimeterHit??

Possible new structure

Data	Current	Proposed	Comment
Cell ID	4 / 8 bytes	4 / 8 bytes	
Amplitude	4 bytes	4 / 8 bytes	High/low gain
Timestamp	4 bytes	4 / 8 bytes	For both BXID and hi-reso TDC
Flag	-	0 / 4 bytes	Gain, Trig bits

Try to add flexibility to store more data depending on subsystems

- LCIO has CHBIT_xxx flags to enable/disable data fields
  - Defining more CHBIT flags to control the extention
  - May not be kept at new key4hep-based systems
  - One possibility is to fix it maximal and rely on compression
- Accessors to use them as word (2 bytes) arrays are useful

Discussions at previous meeting

- LCIO will move to new persistency EDM4hep
  - Not immediate, but not very far
  - Variable data sizes may not be supported
- Variable data size may not be needed if compression works properly
  - Can just take biggest data size
  - Thomas prepared a sandbox to check the issue on data size with compression https://gitlab.desy.de/ilcsoft/calice-bench

Things to do

- Confirm if compression efficiently works
  - With some real data desired – SDHCAL?
- Will agree on final data structure
  - With compression: just take 8x3 + 4 bytes (?)
  - Or a bit of reduction?
    - Cell ID: originally 8 bytes
    - Amplitude: 2+2 might be OK but keep 8 for backup?
    - Timestamp: at least 4+2 so can keep 8 bytes
    - Flags: 4 bytes (or 2 if big difference)
  - Event-wide variables can be kept in the parameter (can be any format)

Schedule

- April-may: finalize the format
  - Maybe a meeting on May
- Finalize the list of variables
- Discussion in technical board?
- Implementation – before summer?
  - Or target earlier – on CERN TB before summer?