Software status for 2nd gen prototypes

Thanks for input from:

Vladik Balagura, Frederic Magniette (SiEcal)
Gerald Grenier (SDHcal)
Katja Krueger (Ahcal)
Katsushige Kotera (ScEcal)

errors & misunderstandings are my own

Daniel Jeans, Aug 2015

DAQ software

Detector
FE-ASIC
DIF
concentrator (LDA, DCC)
raw data

event building
event data

conditions data
database, ...

calibration
calibrated data

online display
events, QC plots

I asked ScECAL, SiECAL, SDHCAL-RPC, AHCAL groups to describe their setup and plans in these categories

The goal is to get an overview of what groups are doing

If combined runs are planned, at least the calibrated data should be easily merged

Using common tools also for intermediate steps may streamline activities and prevent duplication of effort

DAQ software

Detector
FE-ASIC
DIF
concentrator (LDA, DCC)
raw data

event building
event data

conditions data
database, ...

calibration
calibrated data

online display
events, QC plots

silicon ECAL (F. Magniette)

LCIO LCGenericObject (via EUDAQ)

event builder under development
initially as on-line builder
to sample fraction of events
later re-use for full building

nosql database for run conditions
under development

as far as I know, no concrete plan
for storing of calibration constants, geometry

online monitoring based on
online event builder output
root, lcio

DAQ software

Detector
FE-ASIC
DIF
concentrator (LDA, DCC)
raw data

event building
event data

conditions data
database, ...

calibration
calibrated data

online display
events, QC plots

SDHcal (G. Grenier)

LCIO - LCGenericObject

Marlin Processor
→ plan to move to standalone library
(intermediate: RawCalorimeterHit
final: CalorimeterHit)

cell positions in xml files
temperature, pressure,
run config in database

ad-hoc procedure to correct for effects
depending on time from spill start

??

DAQ software

Detector
FE-ASIC
DIF
concentrator (LDA, DCC)
raw data

event building
event data

conditions data
database, ...

calibration
calibrated data

online display
events, QC plots

AHCal, ScEcal: K. Krueger, K. Kotera
generally try to follow 1st gen procedures
ScECAL follows AHCAL procedures

two raw data formats:
LCIO – LCGenericObject
ascii – being phased out: still used for LED calib

MarlinProcessor
simple event building from LCIO or ascii
LCIO – LCGenericObject

using "original" 1st gen CALICE db (I believe)
geometry, light yields, calibrations

adhoc macros/codes/processors determine
calibrations: being streamlined
calibs applied by a few Marlin Processors
CalorimeterHit output

CED used for online display

Summary

Impressive range of solutions...

- some “traditional”, following closely what was done for first generation prototypes
- others based on local people's knowledge and preference

If we/you want to converge to a more common set of tools, it's better to start sooner rather than later

First suggestions: is it feasible to have

- a common event builder library?
- a common database technology and format?
  - it's probably convenient to have separate local DBs for now
- a common event display, online monitor? running on LCIO data

backup

From F. Magniette

We are presently developing an online event building system. This is an event subsampling system that can be distributed on a cluster to increase the percentage of treated events.

It is planned that this system feed a high level online monitoring system. It should also output some common formats : LCIO and Root.

- format/structure of event data
  (LCIO generic object? RawCalorimeterHit? other?)

The core of the event builder will be usable also on files and will generate the previous formats for offline data.

- calibration procedure (from a software point of view)

We have a high level run control that includes plenty of physical procedure like calibrations (threshold, hold time...)

- use of databases: calibrations, positions, and so on

We are presently developing a runDB that collects all the parameters of the system for every run. As it is a nosql database, it is very flexible for making request on any combination of constraints.

From G. Grenier

- event building from raw *ROC data (Marlin processor? standalone? other?)
L'event building est pour le moment fait dans un Marlin Processor.
Nous planifions d'extraire l'event building en une librairie stand-alone qui serait interfaçable avec Marlin processor ou stand-alone application selon les besoins.

- format/structure of event data (LCIO generic object? RawCalorimeterHit? other?)
-- Raw data au format LCIO generic object
-- Format intermédiaire au format RawCalorimeterHit
-- Event after Event building au format CalorimeterHit

- calibration procedure (from a software point of view)
Calibration faite sur le temps depuis le début de spill, run par run.
Procédure ad-hoc pour le moment.

- use of databases: calibrations, positions, and so on.
Description des positions dans des fichiers xml. Possibilités de mettre cela dans une database ultérieurement.
Mesures de températures, pressions dans un database.
Configuration dans une database.
Pour le moment , pas vraiment utilise pour l'event building.

From K. Krueger

the whole software for the 2nd gen prototype tries to follow as much as possible the software for the first generation prototype, so is based on LCIO and Marlin. For some steps this is not fully done yet, but in progress
- event building from raw *ROC data (Marlin processor? standalone? other?)
- for the moment we have 2 raw data formats: text files and LCIO files, before event building. At the moment, for beam runs we write both, for LED runs we still use text files. The goal is to have only raw LCIO files in the future.
- For the text files, we have a Marlin processor to convert the raw text files and to do the event building. At this step there is no calibration or reconstruction, only sorting into events. Similarly, there is a Marlin processor to do the event building for the raw LCIO files (no calibration or reconstruction).
- In the next step, a Marlin processor (actually I think it's more than 1) runs the calibration and reconstruction on the LCIO files after event building to produce fully reconstructed data. Since the LCIO files after event building are identical for raw text and raw LCIO files, this step is the same.
- format/structure of event data (LCIO generic object? RawCalorimeterHit? other?)
raw LCIO file: LCIO generic object
after event building and calibration: CalorimeterHit
for simulation: SimCalorimeterHit
- calibration procedure (from a software point of view)
At the moment, some parts of the calibration procedure are already in Marlin (e.g. selection of MIPs), but some are still stand-alone C++ and root (e.g. LED spectra fitting). We want to move that as much as possible all into Marlin processors.
online display uses Marlin reconstruction and CED
- use of databases: calibrations, positions, and so on
geometry is fully in the database, calibration constants used in the reconstruction (LY, …) as well

From K. Kotera

Current status of EBU layers:
- EUDAC makes LCIO, in the lcio file, the data are LCIO generic object, not RawCalorimeterHit, I think.
- generic objects are reconstructed in run base in a special tool algorithm
  this uses mapping data, calibration data (gain, muon, internal, pedestal) , temperature data, ...
- to date, I think DESY people make calibration using their own processor.
 I heard that they still use text file data for the gain calibration, but it will be going to lcio.
 Regarding EBU, DESY do with HBU. We will make some processors by ourselves too.
- mapping data, calibration data (gain, muon, internal, pedestal) , temperature data, ... we have.
 I wrote only from my poor knowledge. I think it better to ask also Lan and Eldwan in detail for current status.