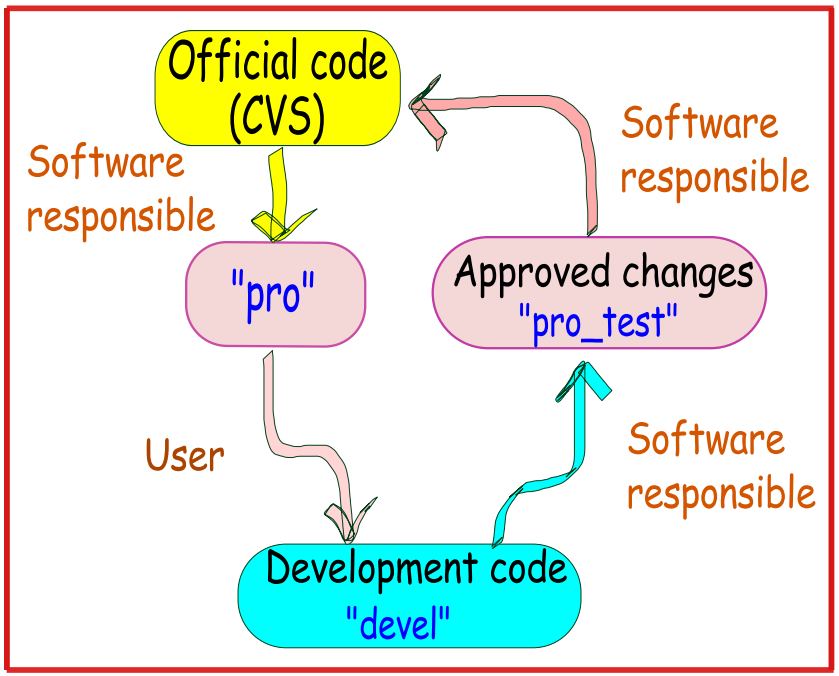
HCAL Reconstruction Status

Angela Lucaci-Timoce

Overview

HCAL Software Strategy

1. Major effort to reorganize our software in the last few months
2. Switched from encapsulated packages (difficult to maintain) to easily manageable central code
3. Build environment: *CMake* (cross-platform and open source)
4. Central *GIT* repository of the HCAL software on AFS
   (*GIT* is a tool for software management, similar to CVS)



5. **Idea**: 3 layers hierarchical structure
   - branch "*pro*": official tagged CVS version, periodically updated via a cron job
   - branch "*pro_test*": "*pro*" + approved modifications
6. Weekly software meeting
7. Advantages:
   - common code used by everyone, updated to the official CALICE software
   - private changes can be fast and easily propagated

HCAL Software from a User's Point of View

1. **BEGINNER**:
   - Initialize: `flcini caliceSoft`
   - Run your job: `caliceMarlin <your steering file>`
   - Get first distributions by using `RootTreeWriter`

2. **ADVANCED**:
   - Detailed instructions on wiki page

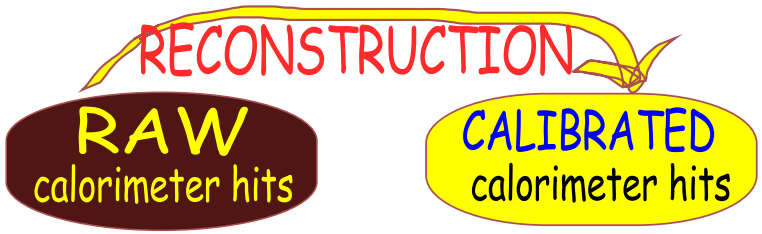
   `http://www-flc.desy.de/flc/flcwiki/HCAL_Software_development`
   - Basic steps: - download the branch with the code you want to modify/improve
     - do your changes
     - tell the software coordinator when you are ready to commit
     - your changes keep your name conserved (useful in case somebody needs details)

3. **USERS outside DESY**:
   - GIT repository publicly accessible:

     /afs/desy.de/group/flc/hcal/calice_soft/git_repo
   - web repository;

     `http://flc-hgf.desy.de/repositories/<repo_name>`

HCAL Reconstruction Steps



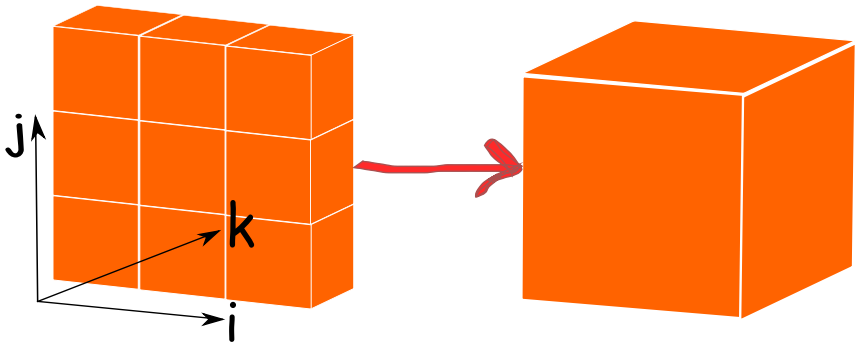
1. Need: acces to the data base information about triggers, slow control, electronics setup and **calibration constants**

2. Calibration constants: averaged over long periods of time (no temperature dependence yet, see tomorrow's talk of Alexander Kaplan)

3. Main **Marlin processor** used in reconstruction:
   fastMappingIProcessor: reads raw calorimeter hits
   maps crate/slot/front end to module/chip/channel
   IntegratedHcalCalibrationProcessor: MIP calibration, zero suppression, non-linearity correction
   fastMappingIIProcessor: reads calibrated hits, maps module/chip/channel into cell ID

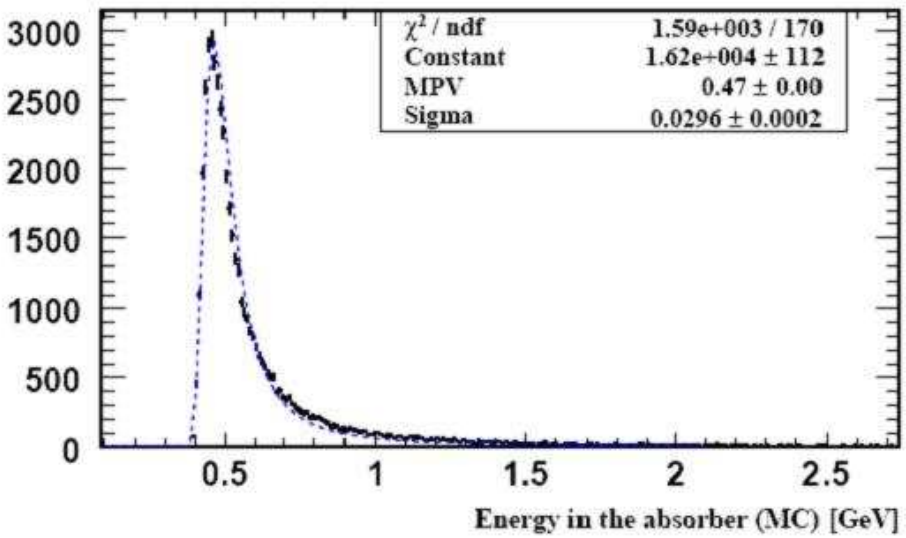
HCAL Digitization

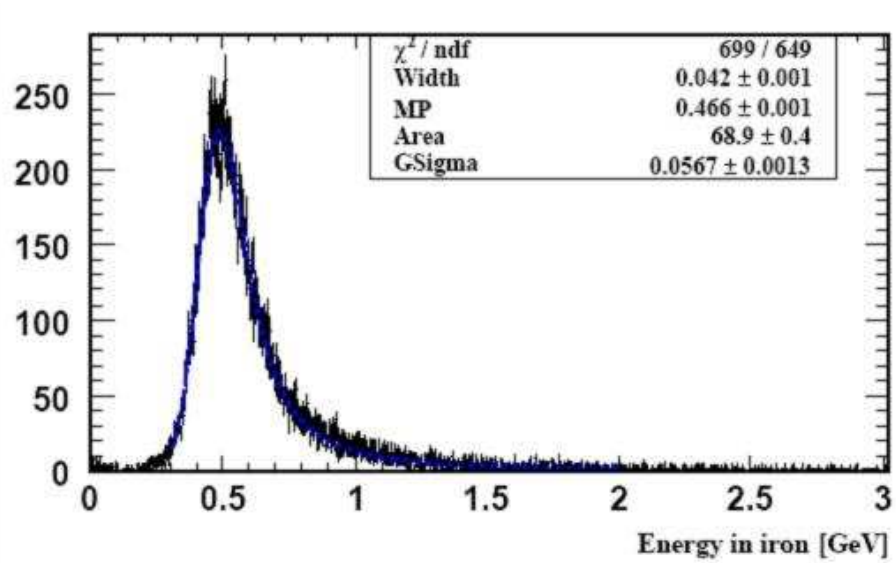
1. **Ganging**: group $1 \times 1\ \mathrm{cm}^2$ Mokka cells into $3 \times 3\ \mathrm{cm}^2$ cells (`ahcalGangingProcessor`)



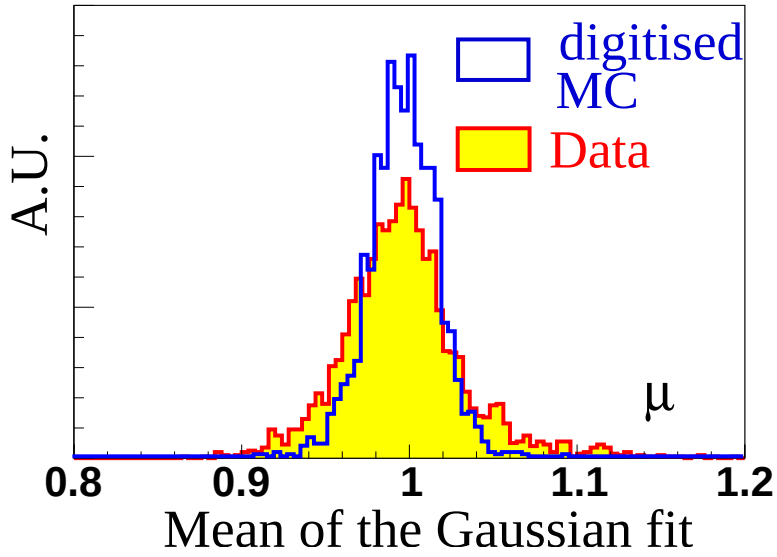
2. Convert geometrical cell ID to hardware information: $i,\ j,\ k \Rightarrow$ module/chip/channel (`fastMappingMCProcessor`)
3. Apply calibrations (`IntegratedHhcalDigitizationProcessor`)

HCAL Digitization: MIP Calibration

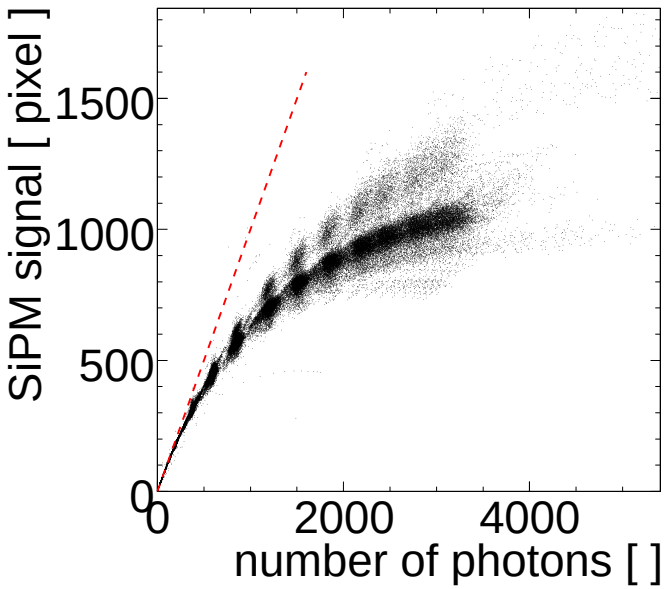




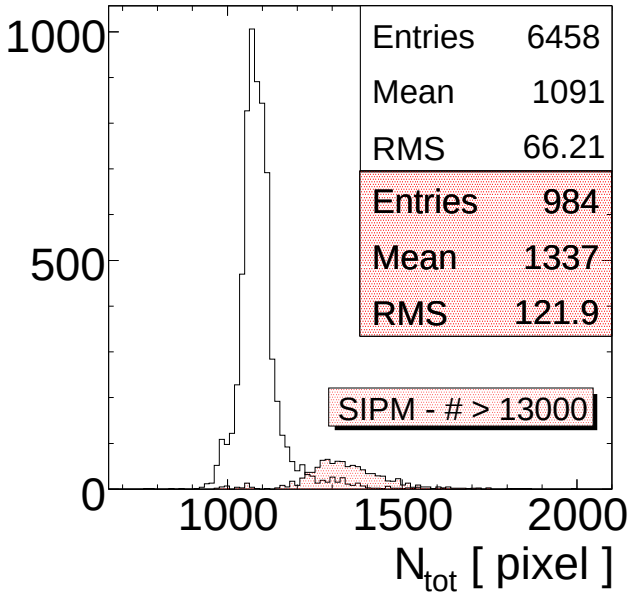
1. Conversion from ADC counts to meaningful energy units (i.e. eV):
   MC: 1 MIP = 860 keV
2. Distribution in data wider than in undigitized Monte Carlo
3. Comparison between 2006 runs and digitized Monte Carlo (**Sebastian Richter**)

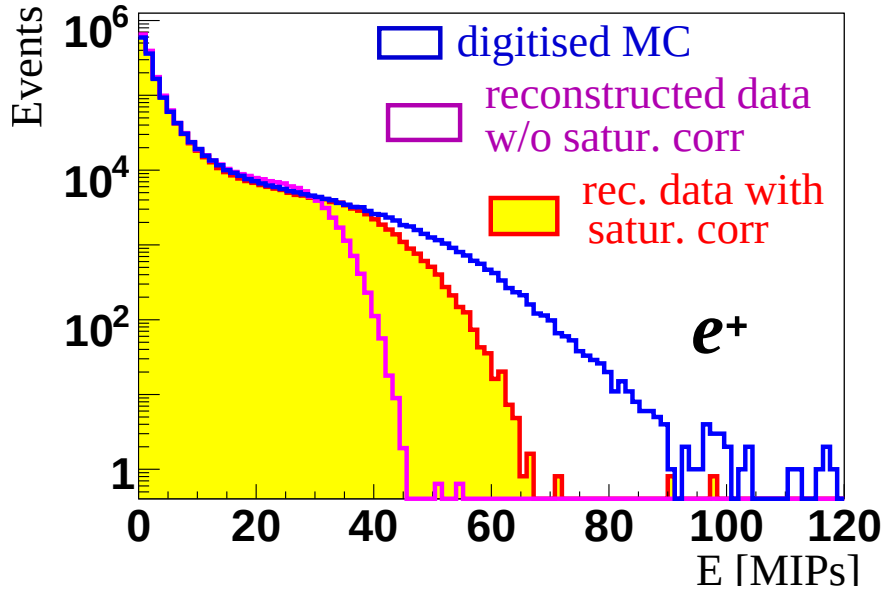


HCAL Digitization: Saturation Corrections

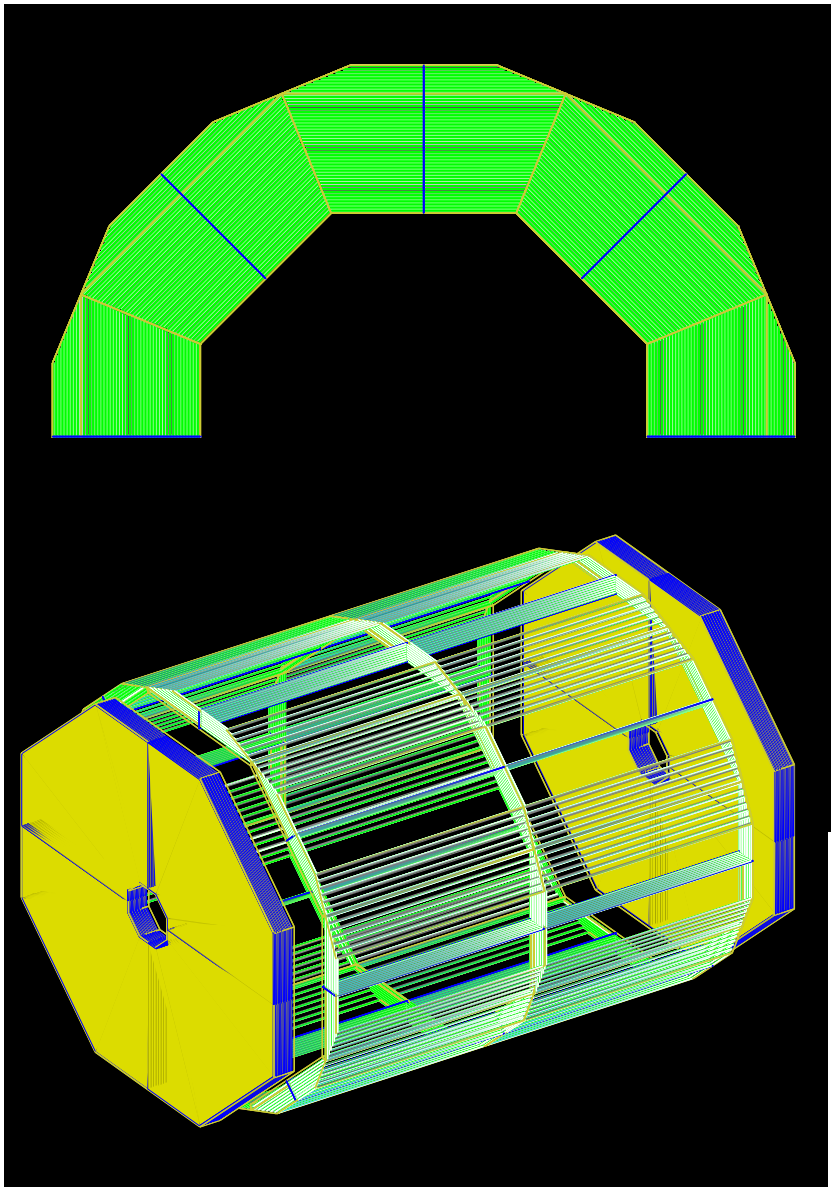


1. SiPM signal $= \sum N_{fired\ pixels}$
2. But: N_{pixels} finite + finite recovery time $\Rightarrow$ saturation behaviour
3. Simple approximation:
$$N_{pixels} = N_{tot} \cdot [1 - \exp(N_{pe}/N_{tot})]$$
4. Saturation curves measured at ITEP (Russia) for every SiPM





HCAL Simulation in Mokka for ILD (I)



1. New HCAL driver, only with **scintillator** option
2. More realistic description

3. **BARREL**:
   - gaps in the middle of a stave
   - layer support structure
   - virtual tiles of $3 \times 3\ \mathrm{cm}^2$
   - **fractional tiles** at the ends of a layer

4. **ENDCAPS and ENDCAP RINGS**:
   - 4 gaps
   - $3 \times 3\ \mathrm{cm}^2$ virtual tiles

HCAL Simulation in Mokka for ILD (II)

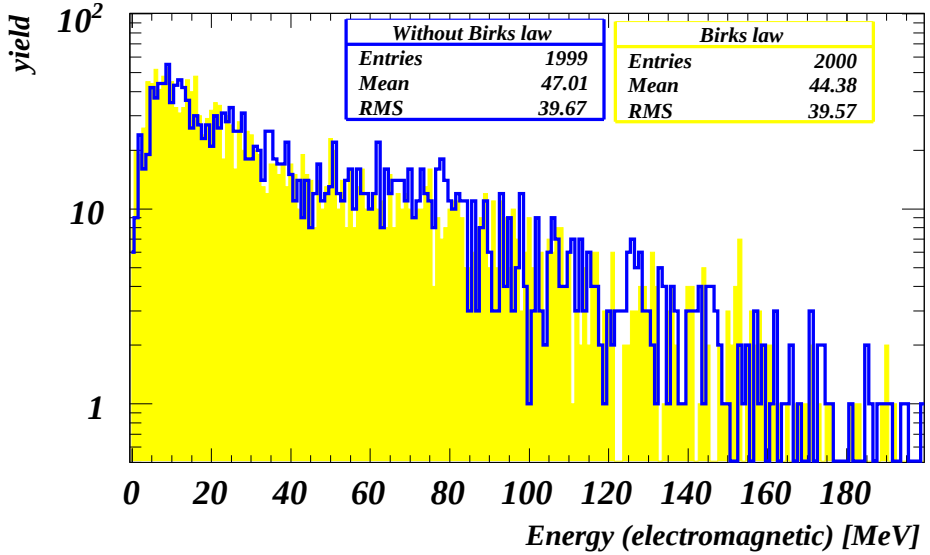
1. **Birks law**: describes the scintillator response ΔL:

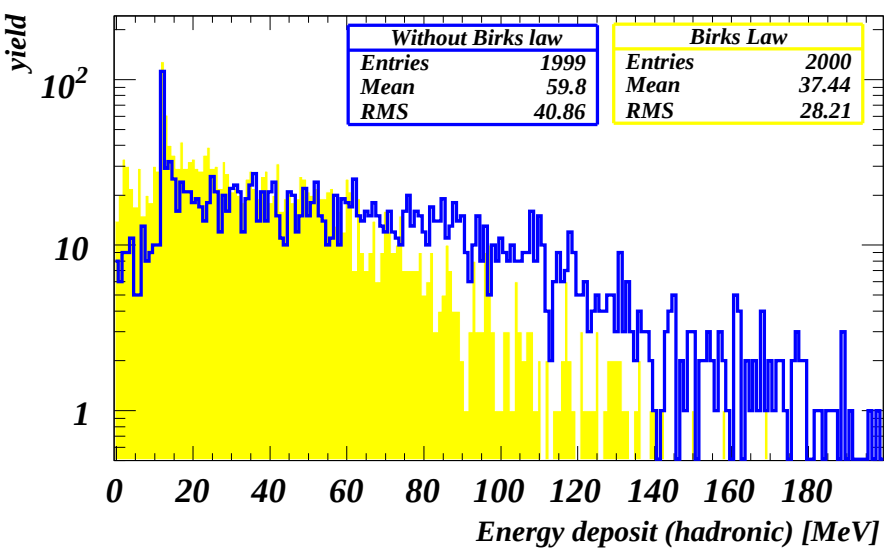
$$\Delta L \propto \frac{\Delta E}{1+k_B \cdot dE/dx}$$

ΔE - total energy deposit
k_B - Birks constant (material dependent);
Polystyrene: $k_B = 0.07943\ \mathrm{mm/MeV}$

2. Implemented in GEANT4 by **Vladimir Ivanchenko** (CERN)

3. Distributions for 60 GeV pions (**Nicola D'Ascenzo**)





Conclusions

1. HCAL software
   - efficient maintainance tools
   - improved documentation: wiki page `http://www-flc.desy.de/flc/flcwiki` and `Doxygen`
2. Reconstruction
   - established procedure
   - MIP calibration understood
   - saturation corrections need more studies
   - work on the calibration constants in the data base finished
3. Digitization
   - first time that we have a working digitization chain
   - calibration effects under study
4. Simulation
   - realistic description of HCAL in **Mokka-06-pre02**
   - model ready for tests production
   - final goal: ILD optimization studies