Report on CALICE Software Model Review

Held at DESY, 18 December 2007

Software Review Committee

Dave Bailey, Paul Dauncey, Gunter Eckerlin, Steve Magill,
George Mavromanolakis, Vishnu Zutshi

Review Subjects and Talks :

- Reconstruction
  - Talk by Roman Poeschl
- Analysis
  - Talks by Niels Meyer and Anne-Marie Magnan
- Database
  - Talk by Roman Poeschl
- Simulation
  - Talk by Nigel Watson
- Management
  - Talk by David Ward
- User's Analysis Experience
  - Talks by Cristina Carloganu and Oliver Wendt

Charge to the Review Committee

- The CALICE collaboration is studying calorimetry for ILC detectors. The collaboration has acquired a large dataset from calorimeter beam tests in 2006 and 2007 and expects to approximately double this during 2008. The total dataset so far is around 300M events, occupying 25TBytes. The dataset has significant complexity, being taken at different locations with differing beam conditions, energies and detectors.
- The ILC detectors have been charged with producing Letters Of Intent by Oct 2008 and initial Engineering Design Reports are expected by 2010. Hence, it is imperative that the collaboration extracts results from these data and publishes them in a timely manner. However, it is also expected that the final analyses of all the data will not be complete until three or four years from now.
- The main aim of the data analysis is fourfold. Firstly, it is to measure the performance of the prototype calorimeters used in the beam tests. Secondly, it is to compare Monte Carlo models with data so as to measure the degree of accuracy of the models. Thirdly, it is to apply the knowledge gained so as to optimise the ILC detector calorimeters with a verified, realistic and trustworthy simulation. Fourthly, it is to develop calorimeter jet reconstruction algorithms and test them on real data as well as simulation.

Charge (cont.)

- A significant offline software structure has already been put together to accomplish these aims, built on a previously determined conceptual model. The purpose of the review is to examine the implementation of this structure and comment on whether it does (or can in future) meet the aims of the collaboration. Some important points are:
  - If missing or ineffective areas can be identified, the review should suggest possible solutions or alternatives.
  - Recommendations to streamline the reconstruction, simulation or analysis of the data, to save effort or time, should be made.
  - The review should examine how well suited is the structure for the connection to the longer term detector studies and the development of jet reconstruction algorithms.
  - Comments on whether the organisational structure is appropriate would be useful.
- There are limited numbers of people involved in the collaboration and so any recommendations from the review need to made with this in mind. In particular, some aspects of the software structure, such as the use of general ILC software, are probably too widely used to be realistically changed at this point. However, as a major user of the central ILC software, our experience should be useful to help improve it. If the review identifies constraints or bottlenecks arising from the use of this central software, comments on these would very welcome.

Overall General Recommendations

Committee realizes that this is a review of "work-in-progress"
-> Review well-defined parts of the software model, pointing out critical items needing decisions, and making recommendations.

General Comments of the Review

-> Documentation and Effort :

Lack of adequate documentation results from insufficient effort available in the collaboration - more CALICE members ouside of the core software group should be involved in production of software - users of software could write usage notes?
A documentation system with an identified leader should be set up asap.

-> Geometry Information :

A common source of geometry information is clearly needed – *unlikely that a central ILC software group would solve this for CALICE.*
A common geometry source should be developed within CALICE for its needs.

-> Use of Central ILC Software :

However, ILC software should be responsive to the needs of users, and CALICE is one of its biggest users – close collaboration is essential to define jobs.
In some cases, it may be more appropriate to use CALICE-specific solutions.

-> Absence of Sci/W ECAL, DHCALs :

Some concern about lack of code for these detectors, especially for Sci/W ECAL for which LCIO-converted raw data exists.

Reconstruction

Recognized as mostly in place -> recommendations were made on how to adapt to changing needs of the collaboration and how to more easily maintain.

-> Steering Files (>1000 lines!) :

Parameters need documentation.

Currently awkward and unwieldy - should consist only of parameters that are different from defaults - easy to spot changes/critical items.

-> Cell/Channel numbering :

Translation between electronic, hardware, geometrical location schemes should be well-defined and transparent – **`MappingAndAlignment`** Classes? Stored in **`cellid`** field?

-> Expertise :

>1 reconstruction expert needed for maintaining and running the reconstruction code.

-> Parameters :

All parameters contained in a steering file must have defaults in a database.

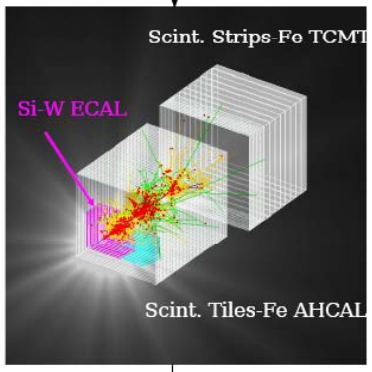
-> Responsibility for detector reconstruction code :

At least 1 person per detector responsible for maintaining (updating, debugging, etc.) code is needed.

Geometry Issues

Geometry Definitions: Mokka database <-> Calice database

Mokka db for Simulation



Might lead to conflicts if simulated setup different from reality

Envisaged step Unification i.e. Feed mokka drivers from calice db

Needed to simulated run conditions beam energy etc.

Calice db for Reconstruction

Roman Pöschl *CALICE Software Review*

Mapping and Alignment - Details

Module Position
ModuleLocation Class
Position of modules
in subdetector coordinate frame

Module Index
arbitrary number
ModuleConnection Class
contains crate, slot, FE layer id
module type

Layer Id* (Largely) Detector independant

Module Type

Module Details
ModuleDescription class
'Cell Shape'
Relation between
cell position in r/o sequence and 'Mokka' index
Position of cell w/i module

Positioning in world coordinate frame by dedicated class DetectorTransformation

Detector independant but hardware dependant

Detector dependant

Scheme invented for SiW Ecal by G. Gaycken and adopted by AHCal and TCMT

*+ indicator to account for vertical subdivision of Ecal

Roman Pöschl *CALICE Software Review* 18

Analysis

Caveat : Most current analyses are being done as part of the reconstruction – typical and standard for test beam analysis - some analyses either require raw data and/or are not stabilized sufficiently to use reconstructed data.

-> Use of raw files is difficult :

Lack of documentation, complexity of database and interface instructions

How can using the raw data be made more transparent?

-> Database access :

Analyses should be able to be performed without access to the main database

Can this be done without requiring extra infrastructure installation?

-> Event display :

No common event display exists – no explicit need expressed -> odd?

-> Full use of data :

90% of users copy data files locally – wastes majority of data, “good” data bias?

Needs central run and event selection list with well-defined standard criteria for good and bad runs – used for all analyses.

-> Common analysis :

Work towards a common analysis high-level structure, e.g., muon id.

-> ILC detector studies :

Consider how to connect test beam studies with detector concept optimization – are there additional technical requirements?

Analysis Issues

Annex A

Software packages needed by

Experts:

calice_reco (+further dependencies), calice_userlib,
(calice_analysis)
LCIO, Marlin, LCCD, CondDBMySQL

Users:

LCIO, calice_userlib (indispensable)

(calice_analysis (recommended))
Marlin (highly recommended)
LCCD, CondDBMySQL
(not needed in first stages of analysis
but cannot be excluded, see later)

Installation of software will be facilitated by application of cmake building environment which become the standard in calice!!!!

Database

-> Data organization :

The organization scheme is complex and not user-friendly! – better documentation and organization of folders is needed. Version control and clearer naming conventions should be adopted. Folder division for parallel data-streams is needed. A cleanup task force should be formed to organize current and future data. A database browser would also be useful.

-> Conditions processor :

Currently uses the Marlin standard `ConditionsProcessor` to interact with the database. Issues are : 1) ALL database folders are opened at initialization (before run header is accessible); 2) it is not possible to make CALICE-specific names for data folders – even for subfolders which are data file independent. Can Marlin be modified to solve these issues? If not, a CALICE-specific processor should be considered which copies the main functionality of the `ConditionsProcessor` and solves these issues.

-> User access :

Several patterns of access to the conditions data are used in the reconstruction and analysis jobs, leading to confusion for new users and non-use of the database in analysis. One access pattern should be selected and appropriate interfaces be written for each set of conditions data, allowing access to the database in a consistent way – allows most flexibility and is usable.

Database Issues

Additional Complications

- Calice

takes data at different Locations CERN, DESY and FNAL (in 2008)
sometimes even parallel
There could have been in principle parallel datataking
of several detectors at the same location

- Marlin

Program execution is piloted by a steering file

⇒different steering files for the reconstruction job
Mainly due to different database folders
Details on database see this afternoon

Conditions Data – Critical Issues

- Starting Point: Nothing else but LCIO needed to work with reco files (+userlib to interpret calice specific data stored in LCGenericObjects), i.e. no LCCD

- ~95% of the conditions data are handled currently in the reco job and hidden from the user (i.e. Non expert)

(As experience and sophistication of analysis grows)
Analysis might require to access conditions data not yet handled

Users have to be ready to use LCCD also
during analysis since it is difficult to predict what might be needed

- There are sets of conditions data useful for analysis, e.g. Dead Cell Map

a) Only Storage in database
b) Identify these data and attach them to the (first) event
in addition to the storage in the database

a) is the cleanest solution (and my proposal)!!!
In any case access to be performed via LCCD interface
Easy benefit from updates of this map

- Studies of different Alignments

Looks like we have to store the difference to the default alignment into the file – Details to be discussed

Database (cont.)

-> Conditions data included in reconstruction files :

Expressed desire is to run analysis jobs without access to database, therefore adding conditions data to the reco files is a reasonable solution. A processor to unpack the conditions data would then make the data source entirely transparent to the analysis processors. Conditions data is accessed only through the standard processor. Example : conditions data already exists in the reconstructed data through `CalorimeterHit` – contains cell central position (alignment conditions data). Ideally, users would not use the positions in `CalorimeterHit`, instead using the processor that contains the alignment information. Requires rewriting much user code. Remove read-only protection and overwrite `CalorimeterHit` when new conditions data appear?

-> Detector concept independence :

LCIO standard allows both Marlin and lcsim.org to read data, however, no interface to the conditions data exists for lcsim.org. CALICE has chosen to do reconstruction with Marlin. The committee does not recommend that code be duplicated for access by lcsim.org.

-> Responsibles :

One person per detector should be identified who will be responsible for maintaining database entries, including documentation and version control.

-> Meta-data access :

Need to form lists in the data base and need tools to access meta-data, e.g., run quality lists.

Conditions Data Issues

Conditions Data - Summary

- Conditions Data are an integral part of the calice data processing
  First experiment withing ILC which produce these data extensively

- LCCD allows to handle conditions data in a clean way
  Common interfaces for different sources

- Users have to be prepared to use LCCD
  Still a major step which users like to circumvent
  Better training, examples needed

- Need to define a strategy to handle conditions data which are intimately needed for analysis or which might need to be changed for systematic studies at the 'core' of the data processing, i.e. During MC simulation

Simulation

-> Geometry :

As already mentioned in the Overall Comments – need a common geometry source.

-> Reconstruction :

Real data and simulation should share as much of reconstruction code as possible. A method to include pedestal corrections (ECAL reconstruction) in the simulation needs to be developed.

-> Misalignments :

Misalignments should be included in the simulation of events.

-> Hadronic shower models :

Review available models and estimate how many runs are required to compare these models with the data.

-> Fluka :

Effort needed to get Fluka simulations of the CALICE data is large – recommendation is that it not be attempted. This recommendation also applies to any simulation package which has the hadronic shower models bound up in the simulation.

-> Book-keeping :

The case for simulation matched run-by-run to the data was presented. Like misalignments, it was felt that run-dependent systematic effects should be modeled in the simulation, maximizing the shower model comparisons - the book-keeping effort to facilitate data-simulation comparisons should not be underestimated.

Simulation Issues

Simulation Model

- Geant4 as simulation framework
- Simulated output in LCIO format, directly comparable with data
- Support for multiple testbeam installations and whole detector models within common framework
- Support for wide range of physics models
- Accessible for grid production and individual users
- Models adaptable for systematic studies

Objectives

Aim 2) "Compare Monte Carlo models with data to measure the degree of accuracy of the models"

- Requires detailed description of all aspects of multiple testbeam installations
  - Physics models
  - Detector geometry/materials/placement
  - Beam profile
  - Digitisation

Aim 3), "Apply knowledge gained from 2) to optimise the ILC detector calorimeters with a verified, realistic and trustworthy simulation"

- Requires
  - The ILC detector concept models to be implemented to the same level of detail/accuracy which is found necessary to obtain acceptable level of agreement with testbeam data
  - Use of same physics models and parameter tunes
  - Prescription to attribute testbeam data-derived uncertainty to predictions of ILC detector concept studies

Management

Management-Lite was presented as adequate and appropriate for CALICE software management, but the committee felt that due to the growth in size and complexity of the data and detector configurations, some well-defined structures and responsibilities are now needed.

-> Identification of responsibles :

As mentioned before, a responsibility structure is urgently needed, with persons identified as responsible for detectors, reconstruction code, database entries, documentation, run quality, etc., ultimately reporting to the Software Coordinator.

-> Collaboration structure :

The roles of the Physics and Analysis Coordinators relative to the Software Coordinator need better definition. Who decides reconstruction run scheduling, coordinates parallel simulation runs, re-reco jobs, etc.?

-> Discussion forum :

Meetings devoted specifically to the discussion of reconstruction and simulation software issues should be held regularly and separately from the “Analysis and Software” meetings which are now held roughly bi-weekly. Reco software sessions should also be scheduled at future collaboration meetings.

Management (cont.)

-> Simulation constants :

Accurate simulations require some constants (e.g., beam spot size) measured from the real beam data. However, sometimes these measurements use simulated results (e.g., the multiple scattering contribution to beam spot size).
The production and reconstruction of data and simulation need to be iterated in a coherent and coordinated way to make analysis more efficient.

-> User base :

The committee expressed concern over the very limited number of people involved in the analysis of the data, compared to the size of the collaboration. It was felt that making the software easier to use, more accessible, and better documented would help in this regard. Analysis of LCIO files is preferred over, e.g., providing ROOT files to beginners.

-> Scheduling :

No plans and schedules about the analysis, goals and priorities were presented.
The committee recommends that a goal-oriented approach should be adopted including deliverables scheduled at fixed dates for results related to detector characterization, performance, and simulation comparisons, etc. Coordination of reconstruction and simulation runs should be scheduled in advance, with deadlines for code and conditions data updates ahead of the planned runs.

User Experience (Cristina Carloganu)

How did we start in Clermont ...

We started end of 2006 to work on testbeam data analysis

Since

- no grid experience
- new in the collaboration
- rec data in the familiar LCIO format
- marlin environment available locally
- the reconstruction code seemed quite complicated
  - *mostly our fault - lack of time or experience*
  - *partly lack of documentation*

work locally on reconstructed data

Conclusion

Working environment, most of the ingredients are there

Suggested software changes:

- re-organise (re-baptize) the code
- add an analysis package
- add a geometry package (unify MOKKA+Calice geometry description)

Suggested organisation changes:

- unify the documentation on a single/interactive(?) web page
- add documentation

We are actually quite pleased with the current software environment ...

User Experience (Oliver Wendt)

'Typical' Analyses using HCAL + TCMT

e.m. studies: need HCAL, plus beam instrumentation, but no ECAL and TCMT

muon studies: need ECAL, HCAL and TCMT (?), plus beam instrumentation

hadron studies: need ECAL, HCAL and TCMT, plus beam instrumentation

- data **and** Monte-Carlo reconstruction/digitisation chain necessary
  - study response, energy resolution, shower profiles, particle separation, ...
  - compare data and Monte-Carlo
- people use **very different** software for these studies
  - official CALICE software
  - 'semi-official' software packages, e.g. HCAL software by Sebastian
  - private software, might be advanced and contain more features, e.g. stand-alone HCAL software by Niels, but also others
- there is no unique way of using **'the'** CALICE software, partly people only 'dump' the information into a ROOT file and do the analysis there
- stick here to the official CALICE software/'semi-official' software package

Conclusions

people are using **heterogeneous** software

- official CALICE software, semi-official versions, private software
- very **difficult** to compare results

- we need more **documentation** and a **central** point to access it
  - perhaps a kind of wiki, where everybody can contribute
  - and/or a portal to collect all the links to the available documentation
  - but this needs maintenance to keep it up-to-date
- usage of **central** reconstruction and simulation/digitasation would **help a lot,** if
  - everybody can access information about the processors and parameters which have been used
  - there is enough documentation ('versioning') to understand what have been done
  - this documentation is kept up-to-date
  - it is still possible to access 'low-level' information to perform studies on the mapping, calibration, ganging, digitisation etc.

Summary of Recommendations

The Committee recognized that the software system and organization as it exists is ~appropriate for the tasks needed. *Much work has been done by highly skilled, innovative, and dedicated people!* As always, however, more effort from more collaborators would clearly be useful.

The most important issues still to be resolved are :

- Documentation is needed in all areas of the software.
- A standardized method of accessing conditions data is needed - should it be contained in the reconstructed files?
- A consistent definition of run and event selection is needed as well as a concise catalogue of runs for analysis.
- The interface between Mokka/Marlin and CALICE-specific software needs to be better-defined and implemented - central ILC software versus CALICE-specific software?
- The database needs better documentation and internal organization.
- The management structure of the software system and its connection to the physics and analysis organization needs definition.