Report on the CALICE Software Model Review, DESY, 18 December 2007

The CALICE Ad Hoc Software Review Committee

March 10, 2008

1 Introduction

A review of the offline software model used by the CALICE collaboration was held on 18 December 2007 at DESY[1]. The review had been requested by the CALICE Technical Board to ensure that the model was appropriate and complete for the needs of the collaboration. The charge of the review is given in Appendix A and the membership of the committee is given in Appendix B. Some terms used in the following have a specific meaning for the purposes of the review and some of these are listed in Appendix C.

2 General comments

The review committee members would like to thank all people who helped prepare the review, in particular the speakers. The talks were extremely helpful and brought out many very important points. In addition, the speakers answered the (at times lengthy) questions clearly and openly, encouraging a very productive discussion. The committee would also particularly like to thank Frank Gaede who accepted our invitation to attend the review; his helpful and perceptive comments were very welcome.

The committee hopes the following lists of issues and conclusions will be helpful to the collaboration. A very important point to make from the beginning is that the software model being reviewed was not complete in all details. The talk by R. Pöschl gives a very good overview of the general status of the model. This makes it clear that there are still parts which need further work and indeed raises many of the issues discussed below. The model is therefore considered to be "work-in-progress", although it is very clear a huge and impressive amount of effort has already gone into the system which exists.

Hence, the committee interpreted its task as follows: to review the parts of the model which are well-defined; to point out the most critical items which still need to be decided; and, for some of these, to make recommendations on these decisions. Issues specific to particular parts of the software system are discussed in each of the following sections. However, some items were either common to all or did not fit into specific areas:

1. Documentation and effort: Almost all areas of the software system seemed to suffer from a lack of documentation. The code has automatic doxygen processing which gives the source code and the comments in it but no further overview or explanation of the structure. The lack of more general documentation is clearly a result of insufficient effort being available throughout the collaboration. The software system is being produced by a small group of people and they often face the problem of only having time to write the code *or* document it, but not both. The committee would like to see more of the collaboration members outside the core software group getting involved in the production of central software. However, this is difficult to do without good documentation of the existing software. To break this loop is difficult but one possibility discussed in the review was to have the users write notes on usage, etc., using Wiki-type pages. This would remove the burden of documenting the basics from the core developers. However, the experts would of course need to be responsible for the more detailed documentation and this is probably needed first. One issue with such pages is that they can become obsolete quite quickly and so this might only be a short-term fix. To build a longer term solution would require the collaboration to find a person

[1] `http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=2427`

who takes responsibility for documentation, and can spot gaps and encourage people to provide the necessary writeups. The committee recommends that some documentation system be set up as soon as possible with an identified leader. Ideally this would also include extra personnel being drafted in to help.

2. Geometry: A common source of geometry information, used by simulation, reconstruction and analysis, is clearly needed. The committee thinks it is unlikely a solution will be ready from the central ILC groups (such as GEAR) on the timescale needed for CALICE. Hence, the committee recommends a solution is developed within CALICE specifically for its needs.

3. Central ILC software: There seem to be several issues resulting from the external ILC software being used (such as the `ConditionsProcessor`, see below) which are being worked around rather than solved at the source. Generally, the ILC software should be responsive to the needs of the users, and CALICE is one of the biggest users. A realistic way of working with the central developers should be found, so that they respond and fix these issues in a timely manner. The central developers are also limited in effort, so working in close collaboration with them and contributing to the central software may be the most productive way to make progress. In some cases it may be more appropriate to use CALICE-specific workarounds; the decision on whether to spend time on fixing problems internally or working with the central software groups depends on the goals and future direction of the central software developers and so again, close collaboration would help in understanding this.

4. Absence of Sci-W ECAL and DHCALs: Neither of these systems was discussed in any detail during the review. This is partially due to not having talks for these systems, although this itself results from the fact that there is no code within the CALICE system for either yet. Indeed this does not seem likely to change in the near future. The committee is concerned about the lack of central code for these systems, in particular for the Sci-W ECAL for which LCIO-converted raw data files exist. It recommends that the Sci-W ECAL group makes a significant effort to integrate fully as soon as possible so as to make analysis of the combined beam tests in 2008 as effective as possible. The DHCALs should also start considering what is needed for integration of the offline analysis of these systems.

3 Reconstruction

The committee recognised that much of the reconstruction software is now in place, as a result of a large and successful effort. Hence, the following are mainly comments to help adapt what exists more to the needs of the collaboration as a whole and to help with maintainability.

1. Steering files: The committee considered the huge steering files (said to be over 1000 lines) needed for the reconstruction jobs were awkward and unwieldy. It recommends that the situation be improved where possible. Ideally, besides the input and output files and the Marlin processors being used, the steering file should only list those parameters which are being set to values which are not their defaults. Indeed, for all standard applications, no such parameter changes should be needed. This would make the steering files substantially shorter and hence make it much easier to spot the critical items within them. In addition, documentation of the parameters is needed if the users are to be able to run the reconstruction with confidence.

2. Channel numbering: Each system needs to refer to channels by their electronic, hardware or geometrical location, depending on the task being performed. The `MappingAndAlignment` classes described may provide this translation and is currently used by the ECAL, AHCAL and TCMT reconstruction although its future use is uncertain. In addition, there are no clear rules for when each scheme should be used; in particular, should the simulation digitisation and reconstruction always use geometric numbering? The committee recommends this area is tightened up and clear guidelines established.

3. Expertise: There is a significant risk in having just one person responsible for the reconstruction phase. It is vital to have at least two more people that will be educated accordingly and be able to maintain, and run the reconstruction chain from start to end, i.e. submit jobs, update code, etc. These persons do not need to be developers themselves; they will act mostly as operators and book-keepers.

4. Parameters: Some parts of the steering files are fed with parameters that are not stored in the database. This is not reproducible and the committee recommends that all the parameters used must be stored in the database. Their default values could be modified by the steering file. However, this should only be used for test purposes; production must use the database values.

5. Responsibles: There is a need of one person per detector (tracker, ECAL, HCAL and TCMT) that will be fully responsible for each reconstruction part, to keep the code up to date, to debug, to liaise with experts in each detector group, etc.

4 Analysis

The committee noted that much of the analysis of the AHCAL data is still being done as part of the reconstruction, running on raw data files. This implies some areas have not yet stabilised sufficiently to be performed directly from the reconstructed files, which means there is not a large amount of experience with issues which may arise when this happens widely. Hence, there is a worry that the following does not cover all the requirements of the experiment.

1. Use of raw files: Most users have as a starting point the reco files. These are more or less enough for first level analysis. But once they want to go back one level and have a look into the raw data then they face significant difficulty. Lack of documentation about, e.g., the hardware setup is one reason. The other is the complexity of the database scheme. Even someone who invests enough time to learn how to use the database interface has difficulty to acquire what is needed, where to look, where to get the latest version, etc. A scheme for transparently migrating to the raw data file, through optionally reading both the raw and reconstructed files in parallel, was discussed but this could take significant effort to implement.

2. Database access: The committee considered that analyses should be able to be performed without access to the main database. Also, this should be done in such a way that no extra infrastructure has to be installed, such as a local snapshot. Clearly, this would mainly be used only for the initial period of an analysis but making this easier would encourage more users.

3. Cell numbering: There is a need for a user to be able to translate between the daq id $\leftrightarrow$ hardware id $\leftrightarrow$ position id numbering schemes. For analysis, the issue is that it would be useful to be able to do this without database access. The `cellid` fields per hit should store this info. If this is not possible then a user interface should be developed so that one can go from one id mapping to the other through the database; this may be the existing `MappingAndAlignment` code mentioned above.

4. Event display: There are various private event display applications but there is no common one for all detectors. It seems odd that no one expressed an explicit need for an event display package. The development of such a package is strongly coupled to the geometry info storage and access mentioned above. The need may become more urgent as more detailed shower reconstruction is performed.

5. Full use of data: Analysis is not generally based on the Grid. 90% of the users (7 out of 8) copy a small number of data files locally and do their analysis. This practically means that users look only into a small fraction of data (about 20 to 50 runs out of about 1000). This raises two issues; firstly the waste of the majority of data taken, and secondly concerns of bias as the runs used may not be typical: indeed they are often chosen as they are thought to be "good" by some qualitative criteria and so may not be representative of the true performance of the calorimeters. The committee recommends a central run and event selection list is maintained, with well-defined criteria for good and bad runs for each subdetector, and that this is used for all analyses.

6. Common analysis: There is currently no common high-level analysis structure, e.g. how a common method of muon identification would be shared between users.

7. ILC detector studies: There were no clear plans presented on how the results of the beam test studies would connect with the detector concept optimisation studies (see the Simulation section also). The committee recommends that this should be considered and the additional technical requirements (if any) be evaluated.

5 Database

1. Data organisation: The database scheme is complex and not user friendly. There is need for better documentation and organization of the folder names and what they contain. At least some version control and clearer naming conventions should be adopted. Specifically, divisions into database folders are needed when different conditions can exist at the same time. This is the case for DESY, CERN and (in future) FNAL data, but also for data and simulation. This exists to some extent but is not documented and does not cover all cases. Hence, it would be useful to have the folders for these divisions defined at the top of the database structure in a way suitable for all systems and then make sure all systems use them consistently. A cleanup task force should be formed to clean the database from obsolete data and organise better current and future entries. The development of a database browser, or more information on any existing ones, would help also.

2. Conditions processor: Currently the software mainly uses the Marlin standard `ConditionsProcessor` to interact with the database. There are two main issues. Firstly, the processor opens all database folders at initialisation time (i.e. before the data file run header is accessible to the processor), whereas with data taking having been done simultaneously at several sites and with both data and MC data files, the correct folder depends on the data file being read. (The folder name organisation recommended above would mean the solution for this problem would be applicable to all conditions data.) Secondly, it is not possible to set CALICE-specific defaults for the data folder names, or more specifically, the subfolder names which are data file independent. The committee recommends that the Marlin developers are asked to provide a processor which overcomes these two problems. If this is not possible on a reasonable timescale, then, although it is not ideal to duplicate code, a CALICE-specific processor should be written which copies the main functionality of the `ConditionsProcessor` but again solves these issues.

3. User access: Following a wide-ranging discussion at the review, it became clear that there are several different patterns being used to access the conditions data in reconstruction and analysis jobs. The committee believes this will lead to confusion for new users and may be (at least in part) a reason for the lack of widespread database use amoung people doing analysis. The committee recommends that one specific pattern is selected as soon as possible and an interface written for each set of conditions data by the relevant responsible people, so as to allow access to the database using this route. The committee believes that the singleton pattern allows the most flexibility and is most likely to be usable in all cases, and hence recommends this be adopted.

4. Conditions data in reco files: Given the above and the desire to be able to run analysis jobs without access to the database, it would seem that adding the conditions data to the reco files would be a reasonable solution. A processor to unpack these data into the conditions singleton would then make the data source entirely transparent to the analysis processors. However, it is essential that the source of the data is hidden from users so that changing between using the database data and the file data is truely transparent. Users directly accessing the conditions data collections in the file should be actively discouraged. One unusual and important case of conditions data in the reco file already exists; the `CalorimeterHits` which contain the cell central positions (which are alignment conditions data). Ideally, users would move away from using the position information in the `CalorimeterHit` to using a singleton containing the alignment information. This would allow them to pick up better alignment as and when it becomes available. However, this would require rewriting a lot of the user code. More realistically, using the new LCIO facility to remove the read-only protection and overwrite the `CalorimeterHit` position in an earlier processor may be a more useful way to proceed.

5. Detector concept independence: While LCIO files can be run through both Marlin and lcsim.org jobs, the infrastructure support for the conditions data does not exist in the latter case. A requirement that all reconstruction be done using only Marlin was stated at the review. This seems reasonable as it requires only the experts to work with Marlin. However, for analysis, the situation will make it harder for SiD members of CALICE to contribute. Most of the database data storage is based on LCIO generic objects and the relevant ones of these would have to be reimplemented in java to be accessed in lcsim.org. This would require a substantial effort and the committee does not consider this to be a realistic solution.

6. Responsibles: There is a need for one person per detector (tracker, ECAL, AHCAL and TCMT) to be identified who will be fully responsible for the database entries per detector. This person will be responsible also for the documentation, and version control.

7. Meta-data access: Tools are needed to give access to meta-data, such as run quality lists. The basic information to construct these lists should be stored in the database and simple tools to access this information should be developed.

6 Simulation

The committee agreed with the statements made in the talk by N. Watson that the comparison of data and simulation is the critical first step to the application of the knowledge gained to the ILC detector optimisations and to the development of PFAs. The test beam should help with more realistic simulation of detector geometries and hadron showers, which in turn should boost confidence in the results obtained from PFAs.

1. Geometry: It is crucial that the method for feeding the correct detector/setup geometry (see also the General Comments and Analysis sections) into the simulation should be worked out thoroughly.

2. Reconstruction: It is clear that the real data and simulation should share as much of the reconstruction chain as possible, both to reduce any possible bias when comparing them as well as to minimise the effort needed for maintenance. Both should give `CalorimeterHit` objects which can be used identically for analysis. However, the existing ECAL reconstruction has a significant amount of the pedestal correction code written to work specifically on the raw data structures of real data and these are not produced in simulation. Hence, the pedestal correction algorithms cannot be reproduced in simulation to check for biases. The committee recommends that this be changed.

3. Misalignments: It is not known how important the effects of misalignments may be. Given this, the software should not be designed to exclude generating and reconstructing misaligned simulation events. This requires some thought in connection with the geometry issues.

4. Hadronic shower models: It was not clear how many hadronic shower models are still considered to be realistic. For those which are, then it will be necessary to generate at least some samples of simulated events with all such models so as to allow a comparison between them. The committee recommends that a review of the available models be done and an estimate made of how many runs (and of which type) would be required to be generated for this purpose. This could have a significant impact on the simulation production needs and so should be evaluated.

5. Fluka: The committee considered that the effort needed to get Fluka running within CALICE would be large. It recommends, due to the many other items requiring work, that this is not attempted unless a major part of the effort of integrating it with GEANT4 into a common interface is done centrally by the GEANT4/Fluka developers. Unfortunately, this seems unlikely to happen on a useful timescale for CALICE. The same conclusion holds for any other simulation package which has the hadronic shower model bound up in the implementation and which cannot be simply interfaced to GEANT4.

6. Book-keeping: To simulate a large number of runs is a major task. In particular, the case for a simulation matched run-by-run to the data was presented, as this would allow detailed modelling of the beam spread, Cherenkov material, noise levels, and bad channels, which can vary significantly run-by-run. The committee feels such a detailed generation may be needed as the detectors should be simulated at a level as realistic as possible for the comparisons with shower models. However, it is concerned that the book-keeping effort should not be underestimated.

7 Management

The committee noted the statements that a lightweight structure was adequate but, in general, the committee considered the management would be more productive if well-defined structures and responsibilities were put in place. No concrete plans, goals and schedules were presented, although the committee would have liked to have seen and commented on all of these.

1. Identification of responsibles: No information was given on the people responsible for the various parts of the software system, in the form of an organogram or in any other format. It appears that there is no explicit organisation of this type. It was also not clear who makes decisions within the structure; e.g. the decision on the date of the next reconstruction run is made by the software or analysis coordinators? The committee recommends that a structure is put in place as a matter or urgency, with a clear line of responsibility. This would include a person identified as responsible for the central code of each of the subsystems (ECAL, AHCAL, TCMT and tracking), for the database, for the documentation, and for other global aspects of the system such as run quality, global alignment, code releases, reconstruction and simulation productions, etc. These people would take directions from, and report to, the Software Coordinator.

2. Collaboration structure: The roles of the Physics and Analysis Coordinators relative to the Software Coordinator also need to be better defined. At present, the roles have become quite mixed with a significant degree of overlap. The issue is where decisions are made on scheduling reconstruction and simulation runs, etc., so as to meet the needs of the analysis deadlines (for conferences and papers). If the three Coordinators are at the same level and all report to the Technical Board, then the latter will have to make such decisions. This would be very different from the usual topics discussed by the Board and so it seems likely a change to the composition of the Board might then be needed. Otherwise, one person needs to be identified who makes such overall decisions which are then implemented by the Software Coordinator and the group below this person. This was not a review of the analysis management but clearly this aspect also needs to be considered when clarifying the responsibilities of each of these positions.

3. Discussion forum: There are regular "Analysis and Software" meetings[2], held roughly every two weeks. However, it appears in practice that these meetings tend to concentrate on analysis almost exclusively, with software being mostly confined to a brief report from the Software Coordinator. People involved in analysis, reconstruction and simulation communicate through private emails, which can lead to very fragmented information. The committee recommends that meetings specifically to discuss software issues are held separately, initially with a roughly similar frequency. This would be the main public forum for discussion by the software group discussed above, although it should be open to any interested member of the collaboration. The committee also recommends that sessions specifically for discussion of software issues are scheduled at future collaboration meetings, separate from the physics analysis meetings, although preferably not in parallel with them as there is clearly a very significant overlap of people.

4. Simulation constants: An accurate simulation requires some constants (such as the beam spot size) to be measured from real data before generation. However, accurate measurements of these parameters can depend on having simulation available (such as to measure the multiple scattering contribution to the beam spot size). Hence, the production and reconstruction of data and simulation need to be iterated in a coherent and coordinated way. This has not been achieved so far and the committee feels this needs to be organised in time for the next simulation production run.

5. User base: It is of concern that only a very limited number of people are involved in the analysis of the data, given the overall size of the collaboration. It was not clear to the committee why this should be the case. However, making the software easier to use, more accessible and better documented can only help in this regard. Providing a set of standard ROOT ntuples for beginners was also suggested, although the discussion raised a worry that this would just make eventual migration to using LCIO files even harder.

6. Scheduling: There were no concrete plans and schedule about the analysis, the goals and the priorities. The committee recommends that a more goal-oriented attitude should be adopted. There should be deliverables scheduled at fixed points over the coming 12 months in terms of results related to detector characterisation, performance, simulation comparisons, etc. Reconstruction and simulation runs should be scheduled well in advance, with deadlines for code and conditions data updates ahead of the runs so the complete system can be tested in time.

[2] `http://nicadd.niu.edu/cdsagenda//displayLevel.php?fid=30`

8 Conclusions

The committee feels the software system which exists is aiming in the right direction. It is clear that a lot of work has been done and there is a lot still to be done. The people involved are highly skilled and innovative, particularly given the uncertain boundary conditions under which they are working. However, more effort from other members of the collaboration would clearly be useful.

The above section list a significant number of items for consideration, of varying degrees of importance. There are several critical issues which must be resolved very soon and the committee strongly encourages the collaboration not to pursue alternatives but to decide on single solutions as soon as possible. The committee believes the most important issues to be:

1. Documentation, which is a significant problem in all areas of the software.
2. The definition of a standard method for accessing conditions data, including the issue of whether to store conditions data in the reconstructed files.
3. The definition and implementation of run and event selection and the cataloguing of runs.
4. The interface between Mokka and the CALICE-specific software.
5. The database internal organisation and documentation.
6. The management structure of the software system and its connection to the physics and analysis management.

The committee hopes some progress will be achieved by the time of the next collaboration meeting in March 2008, where it recommends a session is devoted to discussing how to proceed on these issues.

Finally, the committee hopes this has been a useful exercise for the software group and that they will have some benefit for all the hard work they evidently put into preparing for the review.

Appendix A: 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.

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.

Appendix B: Committee members

The members of the software review committee were

- David Bailey (Manchester)
- Paul Dauncey (Imperial College)
- Günter Eckerlin (DESY)
- Steve Magill (ANL)
- George Mavromanolakis (Cambridge/FNAL)
- Vishnu Zutshi (NIU)

with Paul Dauncey acting as secretary.

Appendix C: Definitions of terms

Some terms are used with a specific meaning for the purposes of the review. In particular:

- "Central ILC code" refers to the packages (LCIO, LCCD, Marlin, Mokka, GEANT4, etc.) which are not specific to CALICE and so are generally written by people outside of the collaboration.
- "Reconstruction" refers to the process of producing reconstruction files from the LCIO-converted raw data files. This also covers studies performed on the raw data files as these tend to be aimed at development of code or constants for the reconstruction itself.
- "Digitisation" refers to processing the output of the simulation, which is in terms of "truth" information (`SimXxxHits`) into data which can be run through (at least part of) the reconstruction.
- "Analysis" refers to the studies performed using the reconstruction files, mainly the final step of producing results aimed for publication.