THE NEW BABAR ANALYSIS MODEL

David N. Brown, Lawrence Berkeley National Lab

Representing the BaBar Computing Group

Abstract

As part of a general Computing Model upgrade, BaBar has deployed a new analysis model. The new analysis model was designed to overcome the major shortcomings of the previous analysis model. In particular, the new analysis model consolidates several redundant data formats, allows users to customize the data format for their analysis, and provides a wider range of data access options than the old model, while maintaining backwards compatibility with BaBar's existing analysis interface and analysis code base. The new analysis model was used in roughly half the analyses submitted to the recent ICHEP 2004 conference, and has been enthusiastically welcomed by the BaBar analysis community.

INTRODUCTION

BaBar is a roughly 600-person experimental High Energy Physics collaboration studying the CKM triangle, CP violation, rare B-meson decays and other physics topics. The BaBar experiment is located on the PEP II collider at the Stanford Linear Accelerator Center (SLAC), and is designed to study decays of the Upsilon-4S resonance, generated through asymmetric e^+e^- annihilation. The BaBar detector consists of a superconducting solenoid with drift chamber and Si tracking, a CsI crystal calorimeter, iron-absorber with muon detection, and a novel particle identification (Pid) device using totally internally reflected Cerenkov radiation. BaBar began taking data in 1999, and to date has recorded roughly 240 fb^{-1} of data.

In late 2001 BaBar concluded that the Computing Model (CM1) being used at that time was incapable of scaling up to the data samples predicted for the near future of the experiment, and that a replacement was needed. A Computing Model upgrade (CM2) was designed and implemented in roughly 1 year, and was deployed starting in fall of 2002. BaBar has now been using CM2 for nearly 2 years, and only legacy support is being provided for CM1.

This paper describes the Analysis Model portion of CM2. First we give the reasons for replacing the old analysis model, and present the goals of the new model. The design of the CM2 analysis model in terms of software and operations are then described. We present the options CM2 provides for physics analysis users to customize reading and writing data. Finally we demonstrate how CM2 has improved the physics productivity of BaBar, and improved the satisfaction of the BaBar analysis community.

THE BABAR ANALYSIS MODEL

For the purposes of this paper we define an analysis model as the software and procedures that allow a user to analyse data to extract physics results. The computing goal of the analysis model is to make that process timely, efficient, and complete. The starting point of the analysis model is the objects produced by reconstruction: tracks, clusters, and other high-level objects, together with their low-level constituents (track hits, cluster crystal energies, etc). The end point is a physics publication or conference presentation.

Analysis begins by assigning particle identities to reconstruction objects, forming particle 'candidate' objects. Candidates representing stable particles are created directly from reconstructed objects, for instance by assigning a pion hypothesis to a track. Candidates representing unstable particles are created by combining several stable particle candidates, such as the 2 pions daughters of a K_s decay. This process may be iterated to produce candidates representing the cascade of decays coming from an initial B meson. These complicated candidates are created using computationally expensive vertexing and combinatorial algorithms.

Candidates are then evaluated in terms of their physical properties such as their mass, momentum, direction, etc. Having selected or otherwise identified useful candidates based on these properties, they are compared statistically to specific physics models in order to extract physical parameters, such as the value of a branching ratio or a coupling constant. It is these final values that are published.

In evaluating the systematic errors on an analysis, it is frequently necessary to examine the selection or creation of candidates, or even the underlying reconstruction objects. Thus the analysis model must retain the connection from final physics results back to the original reconstruction objects they are based on.

Reconstruction is clearly a 'central computing' function, just as the fit to extract a physics parameter is a 'user computing' function. The intermediate parts of the analysis model fall inbetween. For instance, the combinatoric algorithms to create complex candidates may be run as part of a centrally-organized data processing or in a users private analysis. The analysis model must support both common and user computing, and provide a way to coordinate and accommodate the different needs of each.

THE CM1 ANALYSIS MODEL

The original BaBar analysis model was designed before any data were recorded. It was refined continuously until it was replaced by CM2, but its basic design was unchanged.

In CM1 the reconstruction algorithm produced two independent output streams. One stream contained 'traditional' reconstruction objects representing tracks, clusters, and other similar high-level objects, plus the low-level 'hits' that were used to build them. The other stream contained candidate objects created from these reconstruction objects. There were no cross-links between the objects in these streams. A detailed description of the 'reco' stream content can be found in reference [1].

The analysis stream content format was designed by a small number of physicists focused on particular analysis topics, and was therefore somewhat limited in its scope. The 'reco' stream was designed through a large effort of detector physicists. The 'reco' stream was a dead-end for analysis, and no physics publications from BaBar are based on it. Instead, all analyses started with the dedicated analysis stream.

After reconstruction, the CM1 analysis model selected events interesting to particular analyses in a centralized 'skim' production. These 'skimmed' events were written as pointer-collections referencing the original complete event collections. Of order 100 separate skim streams were written, each focused on a small set of physics topics.

In CM1 users performed detailed analysis by dumping the content of the official analysis data format into hbook or root tuples. Roughly 1/2 the tuples started with a particular skim, the others used the complete event collection and made selections on-the-fly. The tuples contained the same candidate information as the official analysis data format, plus additional information that made a particular analysis more convenient and efficient. Analysis tuples were frequentlly huge, of order 1 tera-Byte. Many users wrote programs to reduce the huge tuples to smaller tuples, in order to make their analysis more efficient.

Problems with the CM1 Analysis Model

The analysis model used as part of the original BaBar computing model had many problems. First, the multiplicity of data formats resulted in inefficient use of resources. Having two forms of reconstruction output required more disk space than a combined format, and increased the complexity and fragility of the reconstruction program. The candidate information stored in analysis tuples was redundant with the content of the official analysis data format. Because the tuples were typically larger than the original event, large amounts of storage were used for this duplication. Tuple productions dominated the processing capacity available for BaBar analysis. The heavy job of developing the tuple formats and coordinating the tuple production runs (which took months) prevented the physicists involved from working on analysis or detector issues.

Accessing pointer skims was also problematic. As all the skims referenced the same original data, the servers providing access to the original data were frequently overloaded, reducing analysis efficiency. Since the pointer skims did not add any data content, expensive combinatoric algorithms already run in the original skim selection needed to be re-run when reading the skims. Since skim data were not self-contained, exporting them to sites other than SLAC was difficult, forcing all BaBar users to do their analysis at SLAC.

The complete separation of the analysis and 'reco' output streams meant that the only way to propagate detector-level algorithm or calibration improvements to analysis was to fully reprocess (reconstruct) the data. Similairly, algorithms developed in analysis could not be used in reconstruction without a costly code migration. Finally, it was essentially impossible to study fine detector details in analysis since only candidate properties were available.

The inefficiencies of the CM1 analysis model had real costs in terms of the physics productivity of BaBar. As an example, table 1 describes the aftermath of the discovery by BaBar of a new excited D_s meson. Despite our head-start, BaBar's competitors were able to take over this field, in part because our analysis model wasn't sufficiently flexible to allow us to quickly develop new analyses based on a new discovery.

Table 1: History of the discovery of the $D_s^J(2317)$ and related states.

Event	**Date**
First BaBar Talk on $D_s^J(2317)$ discovery	Feb. 2003
BaBar submits $D_s^J(2317)$ draft to PRL	April 2003
CLEO and Belle confirm $D_s^J(2317)$	June 2003
Belle discovers $B \rightarrow D_s^J(2317)D$	June 2003
Belle submits $B \rightarrow D_s^J(2317)D$ PRL	Oct. 2003
BaBar confirms $B \rightarrow D_s^J(2317)D$	Aug. 2003

Goals for the CM2 Analysis Model design

The goals of the analysis model portion of the BaBar computing upgrade are listed below. These were intended to address the shortcomings of the CM1 model given above.

- Consolidate the reconstruction and analysis formats, to eliminate the duplication and redundancy where practical.
- Support deep-copy skimming, with the depth of copy specified independently for each analysis.

- Allow users to customize the event data output for their analysis by storing their own lists of candidates, including composite candidates, and by adding simple tuple-like data structures of their own design and configuration. This would allow direct use of event data for analysis without first dumping it to a tuple.
- Provide multiple options for accessing production data, allowing users to choose the optimal compromise between speed, detail, and flexibility when reading a collection.
- Support direct interactive access to event data at the root prompt.
- Be fully compatibility with the existing BaBar analysis code library, including the existing analysis interface and framework.
- Deploy the new model without disrupting ongoing analyses still dependent on older data and older software.

THE CM2 ANALYSIS MODEL

In CM2 the reconstruction output has been consolidated into a single nested structure that contains both the 'reco' objects and the analysis candidates. Pointer skims have been replaced by deep-copy skims, where the depth of copy is selected skim-by-skim. Users can customize skim output by adding their own lists of candidates, and by adding blocks of simple values. CM2 still supports the CM1 style ntuple-dump analysis model. It also supports analysis by successive reskim, which affords better support and resource usage. CM2 provides a number of options for accessing data, allowing the user to choose for each analysis the optimal balance of detail and flexibility versus performance. These features are described in detail below.

CM2 data formats

In CM2 The analysis data has been redesigned to hold high-level reco objects (tracks, clusters, etc.) explicitly, with candidates being just a thin shell on top of those. The low-level reco objects (track hits, etc.) are written to a separate stream. The analysis data collections are still self-contained, with references back to the low-level data that can be optionally followed. An explicit example is given in table 2, which compares the data formats used to store clusters in CM1 and CM2.

Candidates are used as scratch space in analyses, so that many copies of equivalent candidate objects may be found in different lists in the event. Direct persistence of candidate objects would therefore be redundant, causing confusion and inefficiency. CM2 avoids this problem by persisting candidate identifier objects, which are derived from candidates through an event-scope factory that insures uniqueness.

Simple candidates in CM2 are stored as lightweight objects containing only references to the underlying reco objects. Composite candidates are stored as references to component candidates, together with the algorithm used to combine or vertex them. On readback, composite candidates are rebuilt recursively, applying the specified algorithm.

CM2 allows users to add basic data structures to the event using a simple interface. Data structures are defined coherently as named UsrData objects, which may be associated either with the entire event or with particular candidates. Content is added to UsrData objects by inserting UsrVariable<*type*> objects, where *type* is float, int, etc. UsrVariable<*type*> objects are identified in the block by a user-supplied name. UsrData blocks may be added to the output stream simply by passing the block name to an output control macro. Persisted blocks are automatically made available on readback.

Table 2: Comparison of the cluster data format in CM1 and CM2. The net size after compression is the same for both formats, however the CM2 format supports more functionality and a more coherent interface.

CM1	CM2
Cluster properties • Calibrated energy • Centroid position • Distance to track • 6 moments	Cluster components • Associated crystals • Energy calibrator • Track reference
Stored in 3 unrelated objects	Stored in 2 nested objects
10 Bytes/cluster	10 Bytes/cluster
Candidate interface	Candidate interface
	Reco cluster interface
	Can apply new calibration
	Can compute new moments

CM2 data access

Framework jobs in CM2 are configured to choose one of several *level-of-detail* (LOD) settings when reading data. *Cache* LOD is the default for analysis. It uses high-level information (such as the parameters of the track fit) stored directly in the analysis data to rebuild reco and candidate objects. *Cache* LOD readback accesses only the analysis stream of the reco or skim output, with a performance similar to CM1 readback. *Refit* LOD rebuilds high-level information from low-level content: ie tracks are refit from their constituent hits. *Refit* LOD readback requires access to the full reco content, either directly or by referencing a borrowed collection. It is roughly an order of magnitude slower than *cache* LOD, but allows new calibrations and alignments to be applied. Other LOD are provided for detector studies, reprocessing, and other specialized purposes.

CM2 data can also be accessed interactively at the root prompt, after loading the BaBar specific schema libraries. Both reco-object and candidate interfaces can be accessed interactively, and references between objects can be followed. Interactive access provides a convenient and fast way to inspect data and perform simple analyses.

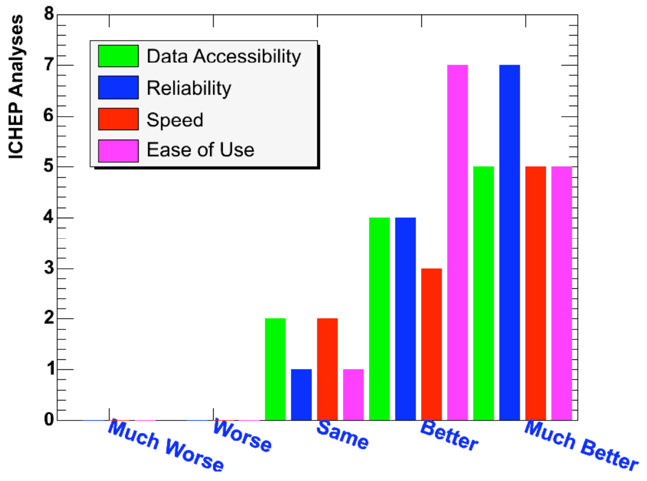
CM2 ANALYSIS MODEL PERFORMANCE

BaBar presented 72 new or updated physics results at the recent ICHEP 2004 conference in Beijing. 30 of these used CM2 for at least part of their analysis. At ICHEP 2004 BaBar presented a paper observing direct CP violation in $B^0 \rightarrow K^+p^-$ using data taken only weeks before [2]. A similar paper published by Belle a few weeks earlier contained only half as much data, and did not have enough significance to claim a signal. It is generally accepted by the BaBar analysis community that this exceptional level of physics analysis productivity could not have been achieved without the benefits of CM2.

Improved access to data with CM2 has helped analyses in several ways. CM2 Skims with persistent candidates can be read roughly 10-times faster than the equivalent CM1 skims. New data are made available to analysis in CM2 much sooner than in CM1. *Refit* LOD has allowed easy estimation of detector-level systematic errors that were virtually impossible to compute in CM1. Figure 1 presents the results of a survey given to users presenting results at ICHEP, indicating a clear preference for CM2.

Providing the reco object interface in CM2 analysis data has facilitated new algorithm development. Because new algorithms in CM2 can provide immediate analysis benefits, the analysis community has become directly involved in their development. For instance, figure 2 shows the improvement obtained after revising the cluster-track matching algorithm to use the space trajectory provided for tracks in CM2. Similar improvements in the MC truth matching, cluster-edge energy correction, and total energy have also been made. BaBar expects significant improvements in data quality once these new algorithms are deployed.

Figure 1: Comparison of CM2 with CM1 by physicists who prepared analyses for ICHEP 2004



CONCLUSIONS

BaBar has deployed a new analysis model as part of its CM2 computing upgrade. The new model corrects the major flaws exposed in our previous model by 3 years of actual use. The new model has already demonstrated significant benefits in the physics productivity of the experiment. The BaBar analysis community has enthusiastically embraced the new model, and are participating actively in its continuing development.

ACKNOWLEDGEMENTS

I would like to thank my co-developers of the CM2 analysis model: Eric Charles (LBNL), Guglielmo Denardo, (INFN Naples), Peter Elmer, (Princeton University), Giuseppe Finocchiaro (INFN Frascati), David Lange (LLNL), and Leif Wilden (NIKHEV Amsterdam). Without their dedicated efforts this project would not have succeeded.

REFERENCES

[1]http://www.slac.stanford.edu/econf/C0303241/proc/papers/TUKT009.PDF
[2] http://arxiv.org/abs/hep-ex/0407057

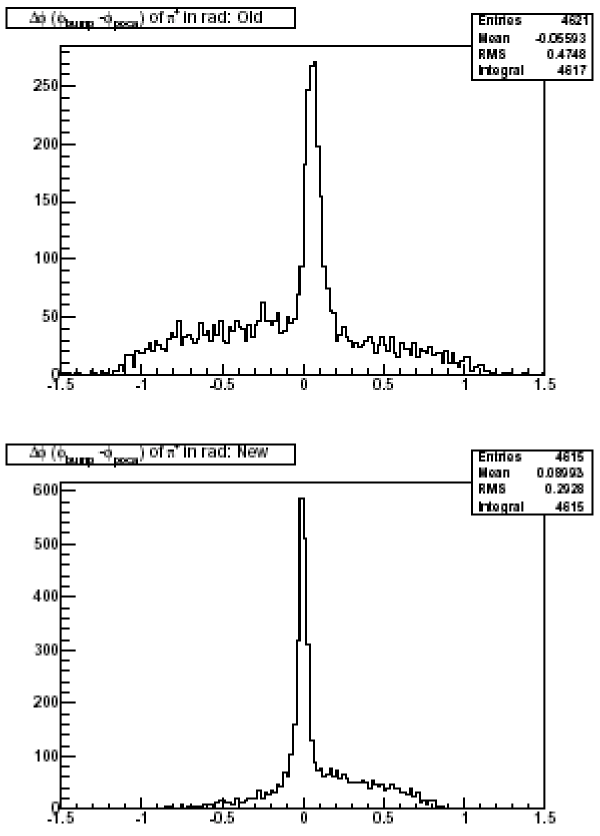


Figure 2: Cluster-track matching ϕ resolution (radians) using CM1 (top) and CM2 (bottom).