

# THE OFFLINE SOFTWARE FOR THE BESIII EXPERIMENT

Weidong Li<sup>1</sup>, Huaimin Liu<sup>1</sup>, Ziyang Deng<sup>1</sup>, Kanglin He<sup>1</sup>, Miao He<sup>1</sup>, Xiaobin Ji<sup>1</sup>, Linli Jiang<sup>4</sup>, Haibo Li<sup>1</sup>, Chunxiu Liu<sup>1</sup>, Qiumei Ma<sup>1</sup>, Xiang Ma<sup>1</sup>, Yajun Mao<sup>2</sup>, Zepu Mao<sup>1</sup>, Jinfa Qiu<sup>1</sup>, Dayong Wang<sup>1</sup>, Liangliang Wang<sup>1</sup>, Shuopin Wen<sup>1</sup>, Linghui Wu<sup>1</sup>, Yuguang Xie<sup>1</sup>, Zhengyun You<sup>2</sup>, Guowei Yu<sup>1</sup>, Ye Yuan<sup>1</sup>, Shilei Zang<sup>1</sup>, Changchun Zhang<sup>1</sup>, Xueyao Zhang<sup>3</sup>, Yao Zhang<sup>3</sup>, Zhi Zheng<sup>1</sup>

<sup>1</sup> *Institute of High Energy Physics, Beijing 100049, P. R. China*

<sup>2</sup> *Peking University, Beijing 100871, P. R. China*

<sup>3</sup> *Shandong University, Jinan 250100, P. R. China*

<sup>4</sup> *University of Science and Technology of China, He Fei 230026, P. R. China*

## Abstract

The BESIII is a general-purpose experiment for studying electron-positron collisions at BEPCII, which is currently under construction at IHEP, Beijing. The BESIII offline software system is built on the Gaudi architecture. This contribution introduces the BESIII specific framework implementation for simulation, calibration, reconstruction and physics analysis. It also describes the implementation of reconstruction algorithms for all sub-detectors and the BESIII event display tool.

## INTRODUCTION

The Beijing Electron Positron Collider II (BEPCII), the upgrade of BEPC, is a multi-bunch  $e^+e^-$  collider that is scheduled to provide collisions in 2007. The BESIII [1] is a general-purpose detector being built to meet requirements of physics studies at BEPCII. The BESIII detector consists of the following components:

- A He gas based Main Drift Chamber (MDC) with a single wire resolution better than 130  $\mu$  m.
- A CsI Electromagnetic Calorimeter (EMC) with an energy resolution better than 2.5% at 1 GeV.
- A Time-of-Flight (TOF) system with a time resolution better than 100 ps.
- A super-conducting solenoid magnet with a field of 1.0 Tesla.
- A RPC based Muon Chamber (MUC) system.

The BESIII Offline Software System (BOSS) is developed using C++ language and object-oriented techniques on the operation system of Scientific Linux CERN (SLC3). The CMT [2] is used as a software configuration tool. The BESIII software uses lots of external HEP libraries including CERNLIB, CLHEP, ROOT [3] etc and also re-uses parts of code from Belle, BaBar, ATLAS and GLAST experiments. The whole data processing and physics analysis software consists of five functional parts: framework, simulation, calibration, reconstruction, and analysis tools.

## FRAMEWORK

The BOSS framework has been developed based on Gaudi [4], which provides standard interfaces for the common software components necessary for data processing and analysis. The overall architecture is shown in Figure 1. The framework employs Gaudi's event data service as the data manager. Reconstruction algorithms can access the raw event data from Transient Data Store (TDS) via the event data service. However, it is the raw data conversion service that is responsible for conversions between persistent raw data and transient raw objects. The detector's material and geometry information are stored in the GDML [5] files. Algorithms can retrieve this information by visiting corresponding services. Through the DST conversion service, the reconstruction results can be written into ROOT files for subsequent physics analyses.

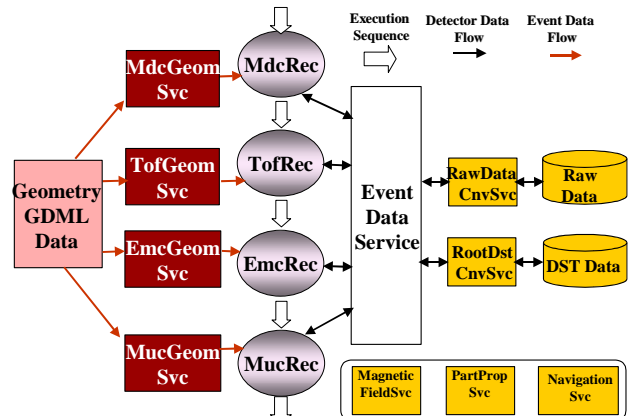


Figure 1: The overall BESIII software architecture.

Furthermore, the BOSS framework also provides abundant services and utilities to satisfy the requirements from different BESIII algorithms. For instance, the magnetic field service provides the value of magnetic field for each space point within the detector. The navigation service will help users to trace reconstructed

tracks back to M.C. particles. Using particle property service, the particles' properties can be accessed by various software components

## SIMULATION

The BESIII Object Oriented Simulation Tool (BOOST), which is based on the Geant4, was originally developed in a stand-alone environment. The material and geometry needed for the detector simulation are read from GDML data files. Due to complicated structure, the standard shapes in Geant4 are not general enough to describe all BESIII sub-detector modules. For example, the G4TwistedTube [6] shape is used to describe MDC stereo cells and G4IrregBox [7] shape is used for EMC end-cap crystals. After importing these two external packages, the BESIII detector can be successfully constructed.

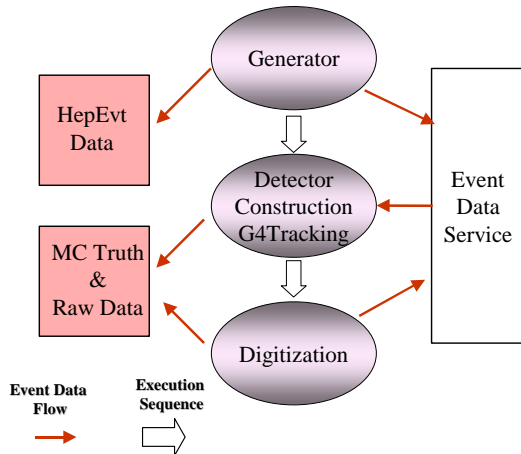


Figure 2: The BESIII event data flow in simulation.

The BESII experiment had about 30 generators written in Fortran language that become BESIII's legacy. In order to integrate the BOOST and legacy generators with the BOSS framework, we have followed Athena's example [8]. As shown in Figure 2, the Gaudi's event data store is used to glue the generator, detector simulation and digitization together. The C++ code uses the HepEvt\_Wrapper class to access the kinematics information of the generated particles, then create a HepMC object containing the kinematics information, and finally register it in Transient Data Store through the Event Data Service. A downstream algorithm retrieves the HepMC object from TDS and passes it to G4Svc [8] for G4 tracking. After digitisation, the raw digit objects will be stored in the TDS for downstream algorithms usage. It is also shown in Figure 2 that the ASCII dump data for HepEvt, MC truth and Raw Data can also be written out from the associated algorithm.

## CALIBRATION

The calibration software, which is based on GLAST's scheme [9], consists of calibration framework and calibration algorithms. The framework provides

reconstruction algorithms a standard way to obtain the calibration data objects. The calibration constants for each sub-detector are produced by the associated calibration algorithm, and then are stored in a ROOT file. The path of the file's location will be recorded in the database along with other useful information like validity of run numbers and the version of calibration dataset etc. When a reconstruction algorithm requests calibration constants, it will trigger the conversion of calibration data objects from the ROOT file. During this process, the framework will first search for the appropriate database record, then fetch only one particular calibration dataset, and finally return transient objects to the reconstruction algorithm.

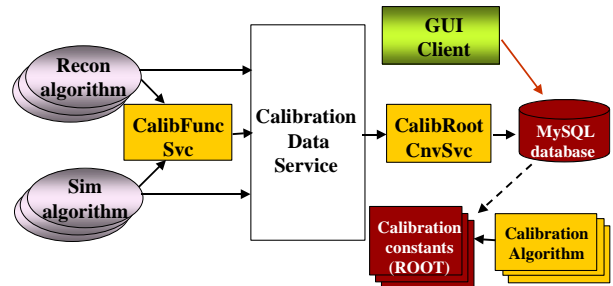


Figure 3: The structure of the BESIII calibration software.

In most cases, both the reconstruction algorithms and simulation need those calibration constants for the same purpose. For example, for the Main Drift Chamber, they will need to convert time to drift distance, to correct the cross-talk effects, to calculate the spatial resolution at certain point inside a detector cell etc. For the maintainability reason, these functions are all wrapped into the CalibFuncSvc service. However, some algorithms still need to directly visit calibration store through calibration data service to retrieve calibration data such as dead channels and channel efficiencies etc. The GUI Client is a suite for calibration operators to handle the administration of calibration database over the web.

## RECONSTRUCTION

Data reconstruction is the central task for the offline data processing. For the BESIII, a complete chain of reconstruction algorithms has been developed and fit into the BOSS framework. The reconstruction system contains the MDC tracking algorithm,  $dE/dx$  and TOF reconstruction algorithms, EMC clustering and shower maker, and muon track finder etc.

The MDC tracking algorithm starts with a formation of track segments from hits using pre-calculated patterns. Then, it links the found axial segments to circular tracks and applies circular fits with a least-square method. After that, it adds stereo segments to track candidates, which is followed by an iterative helix fit. Finally, a track refitting is performed after collecting additional hits in the MDC that might possibly belong to these tracks.

The  $dE/dx$  reconstruction algorithm applies lots of corrections to the ADC values of MDC channels, and calculates precisely the energy loss of each charged

particle through the MDC, then produces the possibility of the particle identification.

A Geant4-based algorithm is developed to extrapolate a MDC track to outer sub-detectors. Considering the magnetic deflection and ionisation loss of the charged particle in the detector, the algorithm provides the precise position and the momentum of the particle in all the outer sub-detectors. The associated error matrix at a given space point is calculated with the multiple scattering effects taken into account. By comparing the results of track extrapolation with the results from a full simulation, it is found that they are well consistent with each other. This track extrapolation is widely used by other reconstruction and analysis algorithms.

For TOF reconstruction, first of all, MDC tracks are extrapolated to TOF and matched with a particular TOF module. Then, the TOF reconstruction algorithm calculates precisely the time that a charged particle needs to travel from the interaction point to the TOF itself after applying various corrections such as effective velocity correction, attenuation correction etc.

The data reconstruction in EMC consists of three concatenated steps. The first step is an energy calibration. The ADC value of each crystal is converted to real energy using the EMC calibration constants. Then a clustering algorithm is called to form EMC clusters for both barrel and end-caps. Finally, the reconstructed showers, with calculated positions and energies, are produced. In this step, seeds are selected using the same criteria in each cluster. The seed is recognized as the local maximum, which has the highest energy among its neighbours. If a cluster only contains a unique seed, a shower is produced at once. Otherwise, a splitting function is invoked to split this cluster into several showers.

The MUC reconstruction algorithm firstly starts two searches for hit collections among axial hit wires and transverse hit wires, respectively. Then it combines two types of hit collections into track candidates and match them with those tracks reconstructed by the MDC. For muon particles with low momentum, they may encounter too few layers in muon counter to form hit collections. In order to improve the tracking efficiency, a second search for tracks will be performed for the remaining hit wires. In this step, MDC tracks will be used as seeds. The seed track is extrapolated to Muon Counter, and hit wires that locate inside a window will be assign to a muon track.

## ANALYSIS TOOLS

The block diagram of analysis software is shown in Figure 4. The Analysis Object Builder will fetch the reconstruction results from TDS through the event data service, then build data objects that are more suitable for physics analysis and put them back to TDS. In this process, the BESIII physics tools are used including vertex finder and particle identification. In BESIII, particle identification will combine TOF,  $dE/dx$  measurements, energy deposits in EMC and MUC information to provide physics analysis particle's IDs.

The kinematics-fitting tool has also been designed and implemented in the BOSS framework. The kinematics fitting can effectively improve the mass and momentum resolution for both charged and neutral tracks.

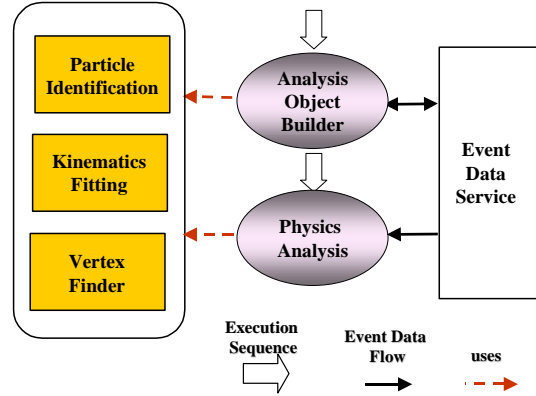


Figure 4: The structure of analysis software.

BESIII event display tool, BesVis, is based on ROOT, XML and GUI. The graphic user interface of BesVis is shown in Figure 5. Detector description information is stored in GDML file, which is also used by simulation and reconstruction. A GDML-ROOT C++ binding interface has been developed to automatically convert XML detector description to standard ROOT geometry objects that will be used by the BesVis to visualize the detector.

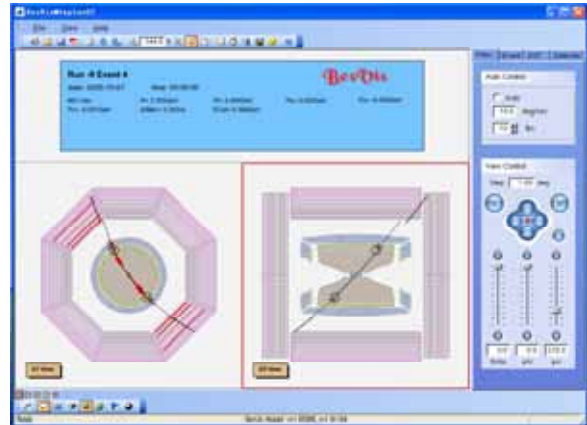


Figure 5: The BESIII event display tool: BesVis.

Both 2D and 3D views are available to satisfy requirements from different users. The GUI based interface is friendly and offers good interactivity between programme execution and users. Raw digits, reconstructed tracks, the association between them and a lot of other relative information could be displayed in an optional way. According to the plan, MC truth information will be added later for software developers to diagnose their algorithms in more effective way.

## CONCLUSION

The Gaudi has been successfully adopted for the BESIII offline data processing and analysis. The BESIII specific framework implementation provides simulation,

calibration, reconstruction and analysis a uniform software development environment. In this framework, a complete offline software system has been developed and currently is being used for detector performance studies and physics preparation. In order to meet the experiment's increasing requirements, the BESIII software may still evolve, however the software's performance tuning will become a long-term task.

### ACKNOWLEDGEMENTS

The BESIII offline software system is part of BEPCII Upgrade Project and is also partly supported by the Knowledge Innovation Project of CAS under Contract Nos. U-602, U-34 (IHEP) and the 100 Talents Programme of CAS under Contract No. U-54.

### REFERENCES

- [1] <http://bes3.ihep.ac.cn/>.
- [2] C. Arnault, "CMT: a Software Configuration Management Tool", Padova, February 2000, Proceeding of CHEP2000
- [3] <http://root.cern.ch/>
- [4] Barrand, G. and others, GAUDI - A software architecture and framework for building HEP data processing applications, *Comput. Phys. Commun.*, 140: 45-55, 2001
- [5] <http://gdml.web.cern.ch/GDML/>
- [6] K. Hoshina et. al., "Development of Geant4 Solid for Stereo Minijet Cells in a Cylindrical Drift Chamber", *Comput. Phys. Commun.* 153:373-391, 2003
- [7] <http://annwm.lbl.gov/G4/G4.cgi>
- [8] <http://www.utdallas.edu/~atescomp/Geant4/>
- [9] <http://www-glast.slac.stanford.edu/software/calib/>