

CMD-3 Detector Offline Software Development

A. Anisenkov¹, F. Ignatov¹, S. Pirogov¹, A. Sibidanov¹, S. Viduk¹, A. Zaytsev^{1,2}

¹ Budker Institute of Nuclear Physics, Novosibirsk, Russia

² E-mail: A.S.Zaytsev@inp.nsk.su

Abstract. CMD-3 is the general purpose cryogenic magnetic detector for VEPP-2000 electron-positron collider, which is being commissioned at Budker Institute of Nuclear Physics (BINP, Novosibirsk, Russia). The main aspects of physical program of the experiment are precision measurements of hadronic cross sections, study of known and search for new vector mesons, study of the $n\bar{n}$ and $p\bar{p}$ production cross sections in the vicinity of the threshold and search for exotic hadrons in the region of center of mass energy below 2 GeV. This contribution gives a general design overview and a status of implementation of CMD-3 offline software for reconstruction, simulation, visualization and storage management. Software design standards for this project are object oriented programming techniques, C++ as a main language, Geant4 as an only simulation tool, Geant4 based detector geometry description, CLHEP library based primary generators, ROOT toolbox as a persistency manager and Scientific Linux as a main platform. The dedicated software development framework (Cmd3Fwk) was implemented in order to be the basic software integration solution and a high level persistency manager. The key features of the framework are modularity, dynamic data processing chain handling according to the XML configuration of reconstruction modules and on-demand data provisioning mechanisms.

1. Introduction

The CMD-3 is the general purpose cryogenic magnetic detector [1, 2] for the VEPP-2000 electron-positron collider [3, 4] which is being commissioned at the Budker Institute of Nuclear Physics (BINP, Novosibirsk, Russia). It consists of a tracking system, barrel and endcap electromagnetic calorimeters based on CsI and BGO crystals respectively, state of the art liquid Xe calorimeter and a muon-range system. The tracking system consists of a drift chamber and two layers of a proportional Z-chamber inside a thin superconductive solenoid with 1.5 T magnetic field. The detailed layout of the CMD-3 detector is shown on Fig. 1.

The main aspects of the physical program of the experiment are the study of known and the search for a new vector mesons, study of the $n\bar{n}$ and $p\bar{p}$ production cross sections in the vicinity of the threshold and search for exotic hadrons in the region of center-of-mass energy below 2 GeV. The VEPP-2000 collider also is going to perform the first test of round beam technique [5, 6].

2. Computing Requirements of the CMD-3 Detector

By design the data acquisition system of CMD-3 detector is capable of running with the trigger rate up to 1 kHz while the average experimental event size is expected to be 3 KB. During the whole lifetime of the experiment (more than 10 years) up to 10 TB of archived raw data is expected to be gathered and stored on tapes and up to 100 TB of online storage space is needed for storing the reconstructed

data, intermediate analysis results and the output of detector simulation system. The batch processing farm hosting at least 50 CPU cores would be required to provide the collaboration with enough computing power for running both reconstruction and detector simulation during the steady data gathering periods.

Compared to the computing requirements of the CMD-2 detector [7, 8] designed for the VEPP-2M collider [9, 10] which was a predecessor of the CMD-3 experiment, the new experiment demands at least an order of magnitude increase of both offline storage system capacity and total computing power accumulated in the offline reconstruction farm. Therefore, despite of the fact that the CMD-3 detector inherited some subsystems from the previous CMD-2 detector, all the components of the data acquisition system, detector simulation tools, and the offline reconstruction tools of the new detector were designed and developed from scratch.

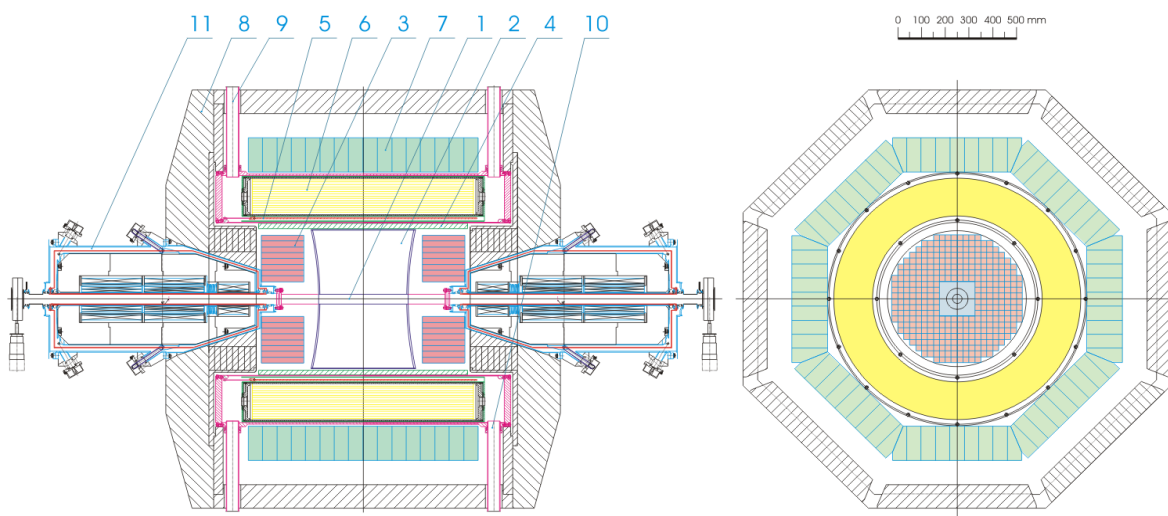


Figure 1. CMD-3 detector layout: 1 – vacuum chamber, 2 – drift chamber (DC), 3 – BGO endcap calorimeter (BGO), 4 – Z-chamber (ZC), 5 – superconducting solenoid, 6 – liquid Xe calorimeter (LXe), 7 – CsI barrel calorimeter (CsI), 8 – iron yoke, 9 – liquid He supply, 10 – vacuum pumpdown, 11 – VEPP-2000 superconducting magnetic lenses.

3. CMD-3 Detector Offline Software Development

3.1. General Design Overview

The main features of CMD-3 offline software design are:

- *Modularity*: an application which deals with the reconstruction of the experimental events within the whole detector or its particular sub-detector is consisting of replaceable and configurable components (modules) bonded in runtime or compilation time.
- *Flexibility*: every module exploited by a reconstruction application is runtime configurable so even changing the sequence of the modules doesn't require recompilation of the application. This feature is normally relied upon during the modules implementation and testing periods, while the pre-compiled module sequences are intended to be used in the production environment. The solution based on pre-compiled sequences is required in order to ensure a reproducibility of the reconstruction environment being exploited during the whole lifetime of the experiment and in addition to simplify migration of the production offline software components from one execution environment to another as the operation system exploited on the reconstruction farm evolves.

- *Data-driven parallelization*: no need to use multiple threads in the event reconstruction applications since splitting the input data and running multiple identical instances of the application (single application per CPU core) is enough.

Most of the offline software developed for the CMD-3 detector is implemented in C and C++ with extensive use of components provided by the STL [11], Boost [12] and CLHEP [13] libraries. Running the reconstruction in a batch mode is assumed to be performed within the runtime environment powered by Scientific Linux 4.x or 5.x (x86 or x86_64 architectures) with the optional XEN based virtualization [14] support.

The important feature of the CMD-3 detector is that it has modular structure which allows one to use any of its sub-detectors for experimental event reconstruction individually. Thus the procedure of the complete reconstruction of the experimental event goes through the following generic stages:

- *Local* reconstruction of the sub-events in every active subsystems of the detector,
- *Global* reconstruction within the entire detector which deals with the output of all the subsystems and produces a high level representation of an experimental event.

3.2. Implementation of the Reconstruction Algorithms

In general, all the subsystems of detector fall into the following categories:

- *Trackers*: Drift Chamber (DC), Z-chamber (ZC), and Liquid Xe Calorimeter (LXe);
- *Calorimeters*: BGO (Bismuth Germanate) crystal based endcap calorimeter (BGO), CsI (Cesium Iodide) crystal based barrel calorimeter (CsI), and LXe Calorimeter;
- *Particle Identification*: Muon and Time of Flight (TOF) subsystems.

All the implementations of hits, tracks, vertexes and clusters are subsystem specific and moreover usually implemented by different groups of developers, though some features are shared among them such as transformations of the object global coordinates within the detector.

Local reconstruction procedure for each detector subsystem implies the following steps:

- *Channel Identifier* (ID) mapping from the *physical* (hardware) ID to the logical ID suitable for the analysis. The transformation is done by the tunable standard module which reads the mapping rules from the calibrations database (ClbrDB),
- Channel amplitudes calibrations and building the object representations for subsystem specific *hits* (calibration data sets are extracted from the ClbrDB),
- Building a compound objects from the hits obtained: *tracks* and *vertexes* for the tracking subsystems (e.g.: DC and ZC) and *clusters* for the calorimeters (e.g.: CsI and BGO detector subsystems).

Since the Liquid Xe calorimeter is capable of measuring both energy deposit and spatial coordinates its reconstruction deals with all four types of objects.

Global reconstruction is intended to compile the information produced by the local reconstruction procedures and group the locally reconstructed objects via exploiting various user defined criteria. It also implies building the higher level object representations such as *global tracks*, *global clusters* and *global vertexes* and in addition performing the high level event identification by producing a set of hypothesis on the nature of the event with likelihood estimators attached.

The hierarchy of reconstructed experimental event components derived from the CMD-3 sub-detectors is shown on Fig. 2.

3.3. Software Integration Platform and Build System

A dedicated software development framework (CMD-3 Software Development and Data Processing Framework, Cmd3Fwk) was implemented in order to be the basic integration solution for building the offline reconstruction code. The design of the framework is based on the following assumptions about data treatment procedure:

- The data analysis sequence is well represented by a directed acyclic graph which has the modules and data instances in the nodes, therefore the reverse call method of building self-organizing modules chain can be used,

- The input data is divided into so called "runs" consisting of so called "events" which have the similar structure within the entire run, thus the cycle over the events and runs for the data set being analyzed can be easily organized by the framework tools themselves, not by some code provided by the user,
- The data instances are produced by the modules and only the creator module of the certain data instance is intended to modify it, so all the intermediate stages of the data processing are preserved.

The key features of the framework are modularity, dynamic data processing chain generation according to the XML configuration of modules and on-demand data provisioning mechanisms. The framework also offers the command-line interface for building XML configurations and running the data processing jobs. All the iterative algorithms are intended to be hidden within the modules, though the interactive module chains can be managed by the framework. In our approach the initial stages of event reconstruction are normally handled exclusively by the offline framework based applications. At the same time the final stages should be handled by implementing interactive or batch mode scripts based on ROOT toolbox [15] or a dedicated Cmd3Fwk module designed for selecting and processing of the particular type of experimental events.

A complete collection of the source code of the reconstruction modules and other components of the CMD-3 offline software is stored in a collaboration Subversion [16] repository and provided with the custom GNU/Make based build system which also interacts with the automatic code generation tools available in ROOT toolbox when needed. The functional layout of the build system is shown on Fig. 3.

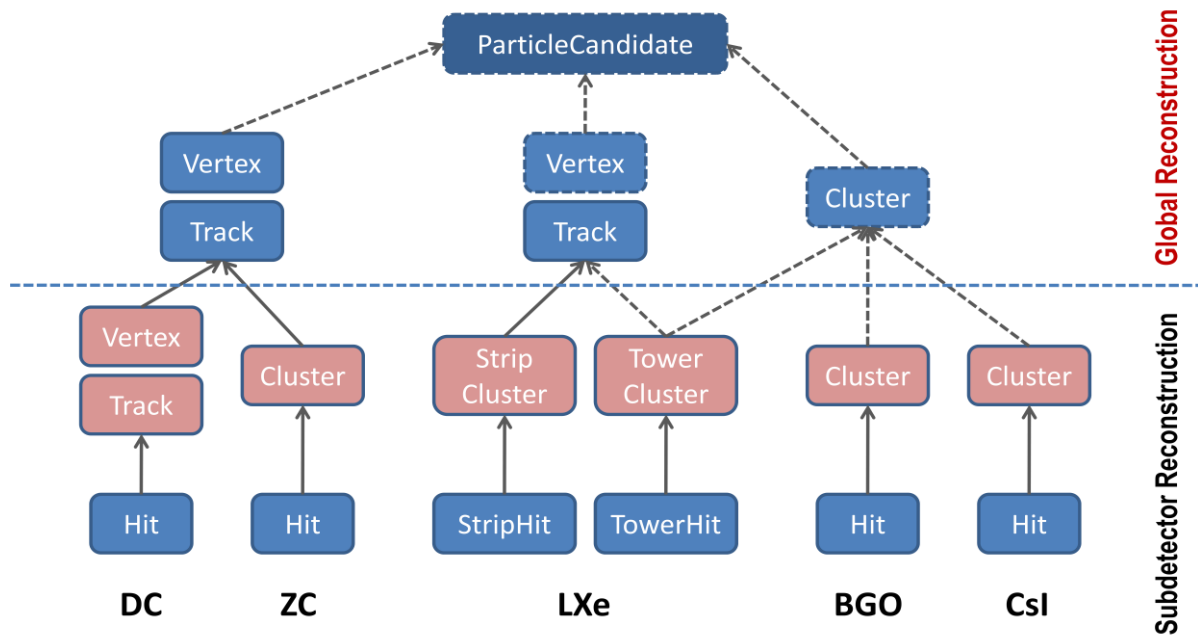


Figure 2. Experimental event reconstruction layout: both local and global stages are shown (solid arrows indicate the parts which are implemented up to the present moment).

3.4. Experimental Event Model and Data Persistency Mechanisms

Since each subsystem of the CMD-3 detector is unique from the point of view of representation of the output produced by the local reconstruction algorithms the high flexibility of the global reconstruction implementation and the availability of an interactive detector geometry and experimental event visualization tools are of crucial importance. The dedicated CmdEvent package was developed to

provide a coherent set of tools for dealing with all the representations of the CMD-3 detector experimental events.

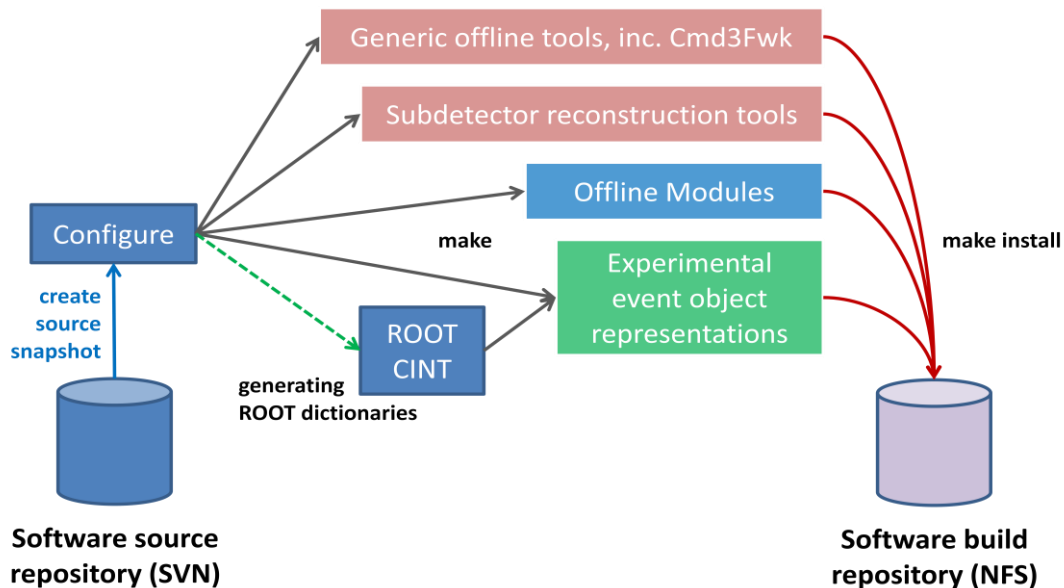


Figure 3. Reconstruction software build system functional layout.

The software package includes:

- Generic containers for the experimental event representation designed to encapsulate an arbitrary set of the reconstructed object collections along with the human-readable names associated with them,
- Persistency management tools based on ROOT toolbox and the correspondent set of data I/O modules for the offline framework.

The tools available in CmdEvent package are capable of representing the following data types being dealt with:

- RAW events which are normally stored in Midas [17] format,
- Output of the primary event generators and Geant4 [18] based detector simulation tools,
- Reconstructed events stored in ROOT TTree containers in ROOT files.

The most commonly used data input/output operations supported by the CMD-3 offline software are summarized on Fig. 4.

3.5. Status of Implementation

Main reconstruction algorithms needed for the CMD-3 detector subsystems are implemented and the work on their improvement and validation is in progress. At the same time many auxiliary components are already in the production state and are extensively used during the test cosmic runs. Recently the prototype of the offline data processing environment was deployed and tested with the data produced by Geant4 detector simulation and experimental data obtained during the CMD-3 test runs. Global reconstruction algorithms are yet to be implemented.

4. Conclusion

The dedicated software development framework (Cmd3Fwk) was implemented in order to be the basic software integration solution for building the CMD-3 detector offline reconstruction applications and running them in a batch mode on the offline computing farm of the experiment. Most of the

reconstruction algorithms and experimental event representation tools needed for detector subsystems are implemented and now being tested and validated. We are now looking forward to intensify the design and development activities related to the global reconstruction algorithms and to apply the tools already available for handling the first experimental data to be obtained by the CMD-3 detector while operating on the VEPP-2000 collider.

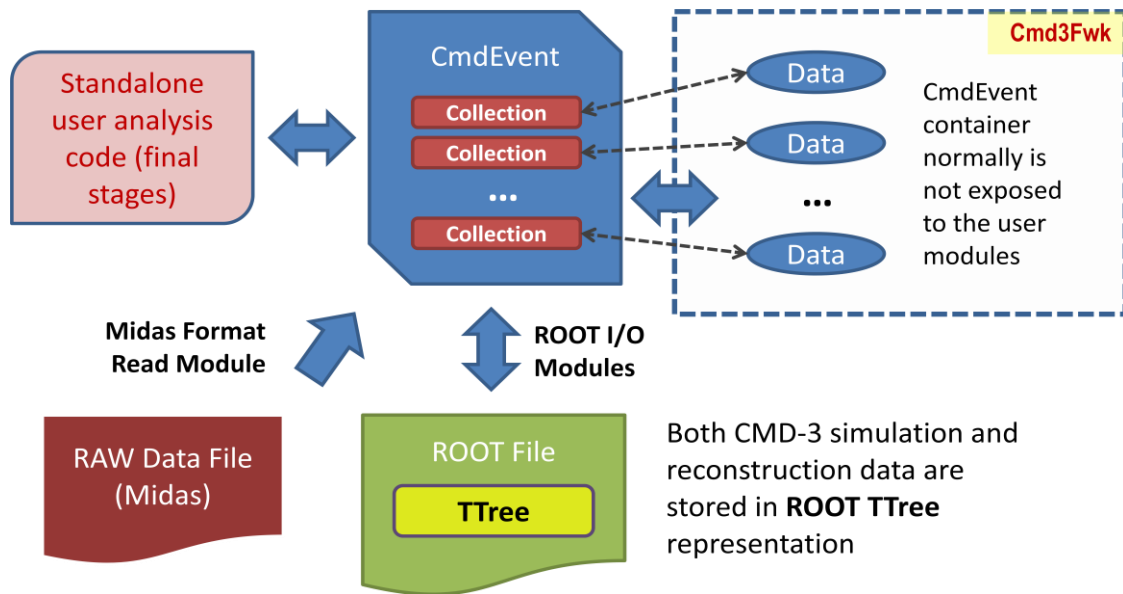


Figure 4. Overview of data input/output mechanisms involved in the offline software.

5. Acknowledgements

The authors are grateful to all the members of CMD-3 online and offline software development groups for fruitful discussions during the design stage and also for useful testing and debugging feedback.

6. References

- [1] D.N. Grigoriev, CMD-2 Detector Upgrade. *Preprint* hep-ex/0106009
- [2] CMD-3 Collaboration Web Site: <http://cmd.inp.nsk.su>
- [3] I.A. Koop, VEPP-2000 Project. *Preprint* physics/0106013
- [4] VEPP-2000 Collaboration Web Site: <http://vepp2k.inp.nsk.su>
- [5] L.M. Barkov *et al.*, Proc. of the IEEE Particle Accelerator Conference, San Francisco (1991), p.183
- [6] V.V. Danilov *et al.*, Proc. of the Asian Particle Accelerator Conference, Tsukuba (1998), p.257
- [7] E.V. Anashkin *et al.*, ICFA Instrumentation Bulletin 5, 18 (1988)
- [8] E.V. Anashkin *et al.*, Instrum. Exp. Tech. 49 (2006) 798
- [9] V.V. Anashin, I.B. Vasserman, V.G. Veshcherevich *et al.*, Preprint INP 84-114, Novosibirsk, 1984.
- [10] VEPP-2M Collider Web Site: <http://www.inp.nsk.su/activity/old/vepp2m>
- [11] Standard Template Library (STL) Documentation: <http://www.sgi.com/tech/stl>
- [12] Boost Library Web Site: <http://www.boost.org>
- [13] CLHEP Library Home Page: <http://cern.ch/clhep>
- [14] XEN Virtualization Platform Web Site: <http://www.xen.org>
- [15] ROOT Data Analysis Toolbox Web Site: <http://root.cern.ch>
- [16] Subversion Web Site: <http://subversion.tigris.org>
- [17] Midas Data Acquisition System Home Page: <https://midas.psi.ch>
- [18] Geant4 Toolkit: <http://cern.ch/geant4>