GEOMETRY DATABASE FOR THE CBM EXPERIMENT AND ITS FIRST APPLICATION TO THE EXPERIMENTS OF THE NICA PROJECT

E.P. Akishina [1], E.I. Alexandrov [1], I.N. Alexandrov [1], I.A. Filozova [1,2,3,a], V. Friese [4], K.V. Gertsenberger [5], V.V. Ivanov [1,6], O.V. Rogachevsky [5]

[1] *Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

[2] *Institute of System Analysis and Management, Dubna State University, 19 Universitetskaya, Dubna, Moscow region, 141982, Russia*

[3] *Plekhanov Russian University of Economics, 36 Stremyanny per., Moscow, 117997, Russia*

[4] *GSI Helmholtzzentrum für Schwerionenforschung GmbH, 1 Planckstraße, 64291 Darmstadt, Germany*

[5] *Laboratory of High Energy Physics, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

[6] *National Research Nuclear University MEPhI, 31 Kashirskoe shosse, Moscow, Russia*

E-mail: [a] fia@jinr.ru

This paper is dedicated to the current state of the Geometry Database (Geometry DB) for the CBM experiment and first results of using the Geometry DB for NICA project. Geometry DB is an information system that supports the CBM geometry. The main aims of Geometry DB are to provide the storage of the CBM geometry, to manage the geometry modules, to assemble various setups as combinations of geometry modules and additional files, to provide its support. The development takes into account the specifics of the workflow for simulation of particles transport through the setup. Both Graphical User Interface (GUI) and Application Programming Interface (API) are available for members of the CBM collaboration. In our approach, the details of the geometry modules are stored in the format of ROOT files. Such a technique allows using the Geometry DB in the NICA project: BM@N and MPD experiments.

Keywords: Geometry Module, Geometry Module, Setup, Geometry Database, DBMS PostgreSQL, CBM experiment, NICA project, Graphical User Interface, Application Programming Interface, ROOT, macros, web-interface.

1. Introduction

The Compressed Baryonic Matter (CBM) experiment is one of major scientific program of Facility for Antiproton and Ion Research (FAIR) in Darmstadt. The goal of the CBM research program is to explore the QCD phase diagram in the region of high baryon densities using high-energy nucleus-nucleus collisions.

Modern scientific studies are characterized by duration, complexity, high labor intensity, large time and financial costs, operating with large amounts of data recorded during the experiment. In this regard, the task of automating the processes of collecting, processing and analyzing the experimental data is of particular relevance. Automation of a modern experiment is impossible without the usage of computer technology, which allows to collect, store and process a huge amount of information, manage the experiment in the process, a large number of experimental equipment at the same time, etc.

Information systems have become an important part of the software of large-scale physical experiments, in particular LHC CERN [1].

2. The brief overview of CBM Geometry Database

The information system for accounting the geometry of the detectors (a geometry database) is intended for storing and processing information about the composition and structure of the detectors used in the sessions of the experiments and physical analysys of the simulated and experimental data. The need for such development is explained in the papers [2-4]. It is necessary to provide flexibility to compile the setups for different sub-detectors: MVD, STS, RICH, TRD, TOF, ECAL, PSD, MUCH, Magnet, BeamPipe. The evolution of geometries should be considered during the facility design works. The same geometry module can be used in the different contexts. Also, the toolkit is needed to manage the geometry modules in a fail-safe, reproducible, and transparent mode.

Thus, Geometry DB as an information system has to provide the following functionality: store the modules of CBM, load the geometry modules for setup construction, construct setup from the stored modules, and support different versions of setup.

The issues dealing with the development of the requirements, the design and implementation of the Geometry DB for CBM experiment are covered in the details in papers [2-4]. The development is performed in accordance to User Requirements Document (URD) [2] that describes the basic concepts *Geometry Module*, *Setup Module* and *Setup*. The *Geometry Module* is the file in ROOT format with the content of the detector geometry. The *Setup Module* is the Geometry module, linked to the mother geometry module (transformation matrix). The *Setup* is the combination of the setup modules and additional elements (the magnetic field and the materials) that describes the full CBM geometry. During the life circle the Setup state may be changed (Figure 1).

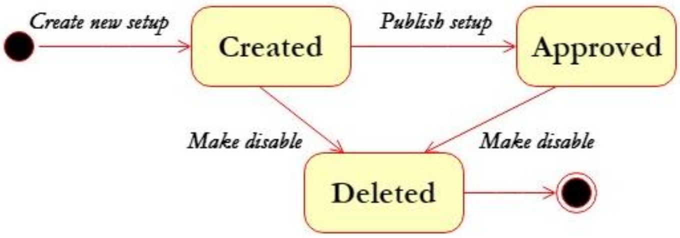


Figure 1. Setup state diagram

The Geometry DB is oriented to three main types of users with different access rights: *Lead Developer*, *Developer* and *CBM user* [2].

The Geometry DB is implemented as a distributed system. The Figure 2 clarifies the deployment of the system artifacts at the nodes. The web-server, database interface, DBMS

PostgreSQL and the origin data are located at the server host. The local database, software CBMROOT and web-browser are placed at the user workstation. The *CBM user* operates only with

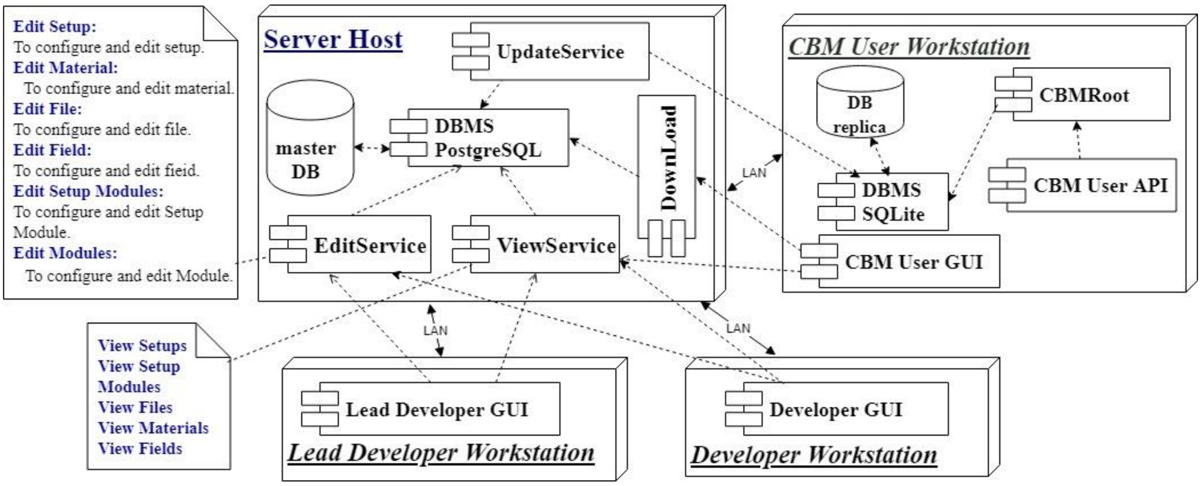


Figure 2. Architecture of the Geometry DB

the local database, downloaded into the framework. The Local Geometry DB (SQLite) [3] is a replica of master Geometry DB and contains only approved setups.

Thus, Geometry DB for storing and retrieving the geometry of CBM modules has been developed. The functionality of the system provides interfaces to view, retrieve and update modules and setups; to store setup as a set of setup modules, magnetic field and materials; to store setup modules as ROOT files and transformation matrix. The realized information system has client-server architecture, involves the database, the web-application (GUI) and a set of ROOT macros (API) [3].

3. Geometry DB adaptation to BM@N

BM@N (Baryonic Matter at Nuclotron) is the first experiment realized at the accelerator complex of NICA-Nuclotron. The experiments CBM and BM@N are very similar especially in their approaches to the methods of the simulation and reconstruction, and also the used scientific software toolkit (FairRoot, FairSoft). The main difference is the set of the detectors. The classes describing the specific experimental detectors are implemented in the frameworks CBMRoot and BMNRoot. The facility BM@N has a greater variety of sub-detectors than CBM has. BM@N collaboration has the actual experimental data allowing to test the Geometry DB.

The details of any subdetector geometry are inside a root file so CBM Geometry DB can be applied to BM@N experiment with minimal changes. Thereby for BM@N the functionality of the web application has been expanded: the function of adding sub-detectors (modules) into the database was implemented, the macros for obtaining of the information about the existing setups and loading the geometry of the selected setup into the memory of the application were updated.

4. Testing of BM@N Geometry DB

First of all, the correctness of the setup geometry obtained from the database for simulation data sets must be guaranteed. To configure the simulation and reconstruction tasks FairRoot package has abstract class FairRun. The FairRunSim is the implementation of FairRun for simulations tasks. The FairRunAna is the implementation of FairRun for reconstruction tasks. The FairRunAna and FairRunSim have different approaches for loading geometry (see Figure 3).

The Reconstraction class loads only the full geometry of facility. The Simulation class load the geometry of different detectors and each detector has a file with geometry. After loading the geometry is stored in a file as a full geometry of facility. This file can be used for reconstraction. The CBM experiment has no real data at this moment, therefore Geometry DB can be verifyed only by the simulation data. The BM@N has real data and geometry DB can be verifyed both: simulation and reconstruction.

One of the main idea of testing Geometry DB is to compare the results of the work of the simulation and reconstruction tasks. The results of the work for the same input parameters should be

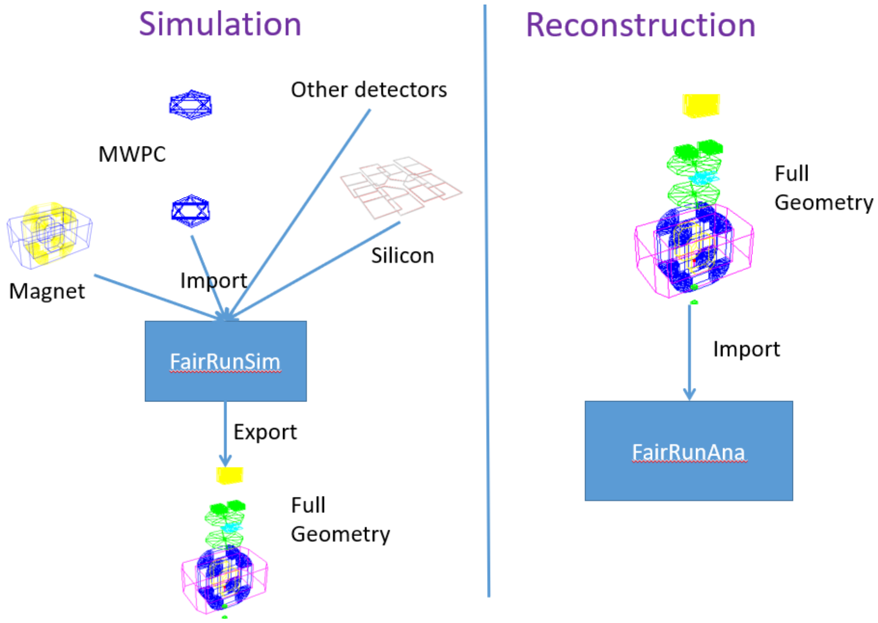
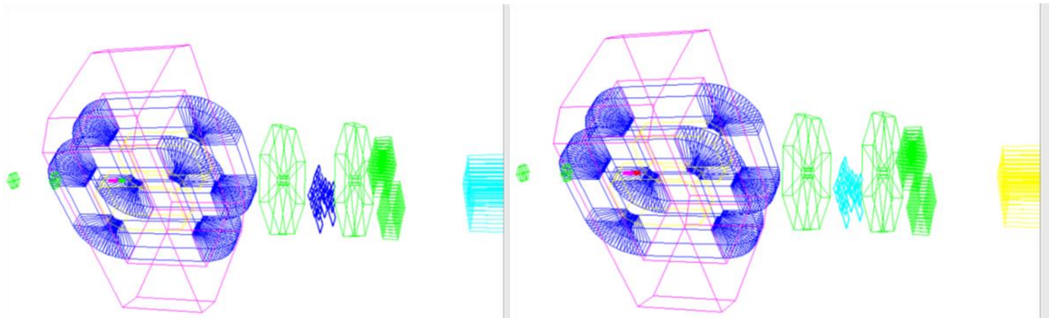


Figure 3. A schema of the geometry loading

the same for two cases: with and without Geometry DB. The simulation task uses the generator of events, so the results data of any new task are different. We can store full geometry of facility which is used for simulation and compare views of them. The Figure 4 presents the full geometry of BM@N after simulation with (b) and without (a) Geometry DB.



(a) (b)

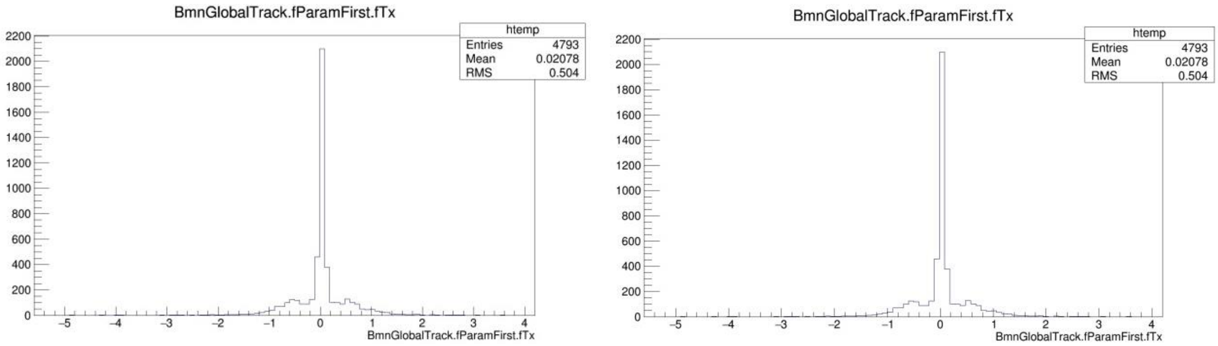
Figure 4. The full geometry of BM@N after simulation with (b) and without (a) Geometry DB.

This way is not very clear because the visual comparison of geometries does not detect small changes (Figure 4). The functional alpha testing was done according to test script (see Table 1).

Table 1. Test script description

	Simulation	Reconstruction
Input Data	Run6, 10 000 events	
Reference to Requirements	Use Case: *Load Geometry* [2-4]	
Expected Result	The results of reconstruction using original geometry and reconstruction from DB with simulation should match.	

Steps	*Test progress*	
1.	To fill Geometry DB with Run6 data	To run reconstruction using the original geometry
2.	To load the setup into the BMNROOT environment	
3.	To run the simulation	To run reconstruction using the geometry from DB
4.	To compare the original geometry and geometry compiled from DB	To compare the reconstructions results
Actual result	The histograms of obtained distributions matched.	



(a) (b)

Figure 5. An example of histograms for the distribution of reconstructed track momenta in Run 6 obtained (a) using original geometry; (b) using geometry from DB with simulation

The Figure 5 presents the obtained distribution of projections of the global track momenta along the X-axis in simulated (a) and reconstructed (b) events of the experiment. The indicators' values *Entries* (number of events), mean (Mean) and RMS (Round Mean Square method) are the same for both cases.

5. Conclusion

The experience of CBM Geometry DB development was applied to BM@N experiment – one of the experiments of the NICA megaproject. The Geometry DB information system for storing and retrieving the geometry of CBM/BM@N modules has been developed. It includes the database, intuitive and compact GUI tools and API tools as a set of ROOT macros. The functional alpha testing was done to demonstrate the correctness of building the setup on the simulation data set.

References

[1] D. Barberis. MODERN SQL AND NOSQL DATABASE TECHNOLOGIES FOR THE ATLAS EXPERIMENT //CEUR Workshop Proceedings. - 2017, Vol-2023, pp.15-22

[2] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov, User Requirements Document of the Geometry DB for the CBM experiment, electronic publishing. Available at: http://lt-jds.jinr.ru/record/69336?ln=en (accessed 30.10.2018).

[3] Akishina E. P. a.o. Development of the Geometry Database for the CBM Experiment / Akishina E. P., Alexandrov E. I., Alexandrov I. N., Filozova I. A., Friese V., Ivanov V. V. // Particles and Nuclei, Letters, 2018, Issue 1, pp. .97-106.

[4] E. Akishina et al. Status of the geometry database for the CBM experiment/E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov//CEUR Workshop Proceedings. - 2017, Vol-2023, pp.144-154.