

Development of the Geometry Database for the BM@N Experiment of the NICA Project

Elena Akishina¹, Evgeny Alexandrov¹, Igor Alexandrov^{1,*}, Irina Filozova^{1,2,3}, Konstantin Gertsenberger⁴, Victor Ivanov^{1,5}, Daria Priakhina¹, and Galina Shestakova¹

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

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

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

⁴Laboratory of High Energy Physics, Joint Institute for Nuclear Research, Joliot-Curie 6, 141980 Dubna, Moscow region, Russia

⁵National Research Nuclear University "MEPhI", Kashirskoe shosse 31, 115409 Moscow, Russia

Abstract. The article is dedicated to the current state of the *Geometry Database* for the BM@N experiment of the NICA project. The main goal of the database is to provide a central storage of the BM@N geometries, convenient tools for managing its geometry modules, various software assembling versions of the BM@N setup as a combination of modules and additional files providing support for a set of versions. Both the Graphical User Interface (GUI) and the Application Programming Interface (API) have been developed.

1 Introduction

The NICA (Nuclotron-based Ion Collider fAcility) project of the Joint Institute for Nuclear Research (JINR) in Dubna aims to study hot and dense QCD matter in heavy ion collisions in the energy range $\sqrt{s_{NN}} = 4 \div 11$ GeV. The rich heavy-ion physics program will be performed at two experiments: BM@N (Baryonic Matter at Nuclotron) at beams extracted from the Nuclotron and MPD (Multi-Purpose Detector) at the NICA collider [1]. BM@N [2] is a fixed target experiment being the first one of the NICA megaproject. The experimental setup was installed in the fixed-target hall, where particle beams from the Nuclotron are extracted, to perform the research program focused on the production of strange matter in heavy-ion collisions at beam energies from 2 to 6 AGeV.

The BM@N experiment was started in 2015, and a set of technical runs was performed with deuteron, carbon, argon and krypton beams collided with various targets, from polyethylene to lead. The analysis of the recorded data is in progress. The nearest plans are associated with an upgrade, so that the experiment could be conducted not only with beams of medium nuclei, but also with heavy nuclei up to gold, and at a higher interaction rate. The upgrade of the experiment entails upgrades of the software systems including the software that describes the BM@N geometry.

*e-mail: alexand@jinr.ru

2 Motivation and guidelines of the *Geometry Database* development

The BM@N experiment should record particles produced in collisions with high efficiency and estimate the parameters with high precision for full study of the produced matter, that is why BM@N consists of sub detectors of different types, strategically positioned around a fixed target, to combine the high precision track measurement with time-of-flight data for particle identification and the total energy measurement for event characterization. The BM@N geometry is required for event data processing, such as simulation and reconstruction in the BM@N software, i.e. *BmnRoot* [3] based on the *FairRoot* [4] package, which was developed at the GSI Institute (Darmstadt, Germany). Users can choose a predefined BM@N setup as a set of modules corresponding to different sub detectors. This procedure is realized via *ROOT* macros. The macro loading the full setup geometry should take into account the geometry hierarchy. The position of modules in the BM@N setup is defined by means of transformation matrices. The matrix is a part of the *ROOT* geometry file, but sometimes it is necessary to move detectors of the BM@N setup on the fly (during the geometry loading) without changing the *ROOT* file. Currently in this case users should change *ROOT* files for such temporary transformation. In addition, users must know the correct geometry macro for a necessary setup. The BM@N geometry should be the same when a user wants to compare results of analyses of Monte Carlo and experimental data. Therefore, it is important for any version of the BM@N setup to store its geometry both in one *ROOT* file and as a set of corresponding versions of modules, each of which is contained in a separate *ROOT* file. The full BM@N geometry setup is stored in a file of the *ROOT* format, and the modules can be stored in separate *ROOT* files as well; this is convenient for simulation goals when detectors are under construction and can be changed. Furthermore, a set of detectors in the setup can be different for different setup versions. At present, the geometry files are stored in the *BmnRoot* repository in many versions (described by revision numbers), and in some cases the name of the class for the same module can differ in different versions of the module. It is also necessary to take into account that some revisions of different modules are not compatible and can not be joined in one BM@N setup.

All of the above show that a geometry database should be created with the following design guidelines: the geometry database should manage module geometries as *ROOT* binary objects; for each module it is required to keep a tag, a version, a transformation matrix and a mother module; pre-defined setups are defined as combinations of module geometries and managed by the geometry database; the geometry database should manage module and setup versions; *ROOT* macros should be implemented to load BM@N geometries from the geometry database; the geometry database should provide low response time and high availability for read operations.

3 Object model and general architecture of the *Geometry Database*

The developed object model of the *Geometry Database* is shown in Figure 1. Any object in the *Geometry Database* has four parameters: *tag*, *date*, *description* and *author*. The *tag* is a specific string, which provides a link to the corresponding object in the database. The *date* is the time of the object creation, the *description* parameter is a short description of the object, and the *author* is the user who created the object. The URL parameter is a string with the full path to the data file, and most of the objects in the database have this parameter. The *Setup* object has a specific parameter *status* that can have three possible values: “Created”, “Approved” and “Deleted”. The status “Created” is assigned after inserting a new setup record into the database, “Approved” – after the action approval by the responsible person, “Deleted” – after the action delete, respectively. The delete procedure means to make this setup unavailable for usage. The setup can be used by BM@N users when its status has the value “Approved”. The *Setup* has associations with one *Materials List*, one *Magnetic Field* and several *Setup Module* objects. The *Setup Module* object represents any detector of the

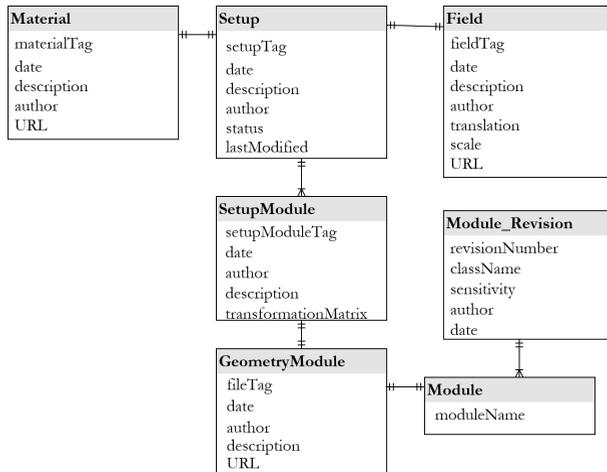


Figure 1. Object model scheme of the *Geometry Database*

BM@N experiment. This object has two parameters: *ParentModule* and *TransformationMatrix*. The parameter *ParentModule* is a tag pointing to the parent setup module, which corresponds to the mother detector. *TransformationMatrix* is used to calculate the coordinates of the setup module in the coordinate system of the parent module. The *Setup Module* is associated with one *Geometry Module* object with the URL parameter linking to a file with the geometry of the detector in the ROOT format. The *Module* object represents a detector and parameter “module name” is the name of the detector. The *Module* object is associated with one *Module_Revision* object. The *Module_Revision* describes C++ class which used for detector geometry loading. It keeps the revision number, the name of this class and special parameter “sensivity” that is used in constructor. These data are used in the API macro for module initialization and BM@N geometry loading.

The general architecture of the developed BM@N *Geometry Database* is presented in Figure 2. The BM@N *Geometry Database* is based on the general architecture of the *CBM Database* [5, 6].

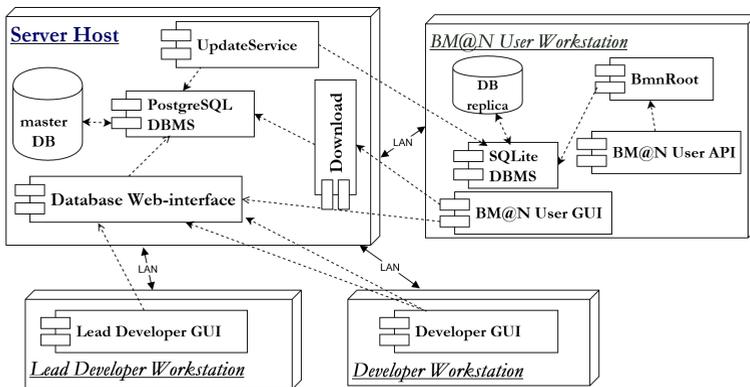


Figure 2. General architecture of the *Geometry Database*

The BM@N *Geometry Database* consists of the following two parts: *Central* and *Local Geometry Databases*. The *Central Geometry Database* provides all the functionality for data

file manipulation and has been implemented at the server host under Apache server with PHP scripts. Lead developers update the *Central Geometry Database* with data files corresponding to the objects of the *BM@N Geometry Database*. To solve the task, both GUI and API interfaces can be used. The *Local Geometry Database* is part of the *BmnRoot* framework and has been implemented on the basis of the *SQLite* database management system as a local replica of the central *PostgreSQL* database.

4 Implementation of API and GUI database interfaces

To operate with the *BM@N Geometry Database*, two tools have been implemented: *ROOT* macros for loading geometries in the *BmnRoot* simulation and reconstruction macros and the Web interface to retrieve the given module or setup geometries. The Application Program Interface has been implemented as *ROOT* macros of the *BmnRoot* framework. There are two main macros: *getSetupList* prints a list of available *BM@N* setups, and *loadSetup* loads the *BM@N* setup into the *ROOT* environment. The necessary setup name and the path to the XML file are used as input parameters of the function. The XML file describes additional transformations of geometries loaded for each module. The load setup macro has been developed once for all versions of geometry modules. It creates modules of the *BM@N* setup (*FairModule* interface) using the class name of the module.

The GUI (Graphical User Interface) has been implemented as a convenient Web-interface, which provides the following possibilities: adding setups, magnetic fields, materials, setup modules and geometry modules; deleting, approving and modifying *BM@N* setups; downloading setups from the Web-interface; producing the replica of the *Central Geometry Database*; viewing setups, their attributes and other details.

5 Conclusions and perspectives

The usefulness of the geometry database and its scope have been shown. A workable prototype of the *Geometry Database* has been developed. The *Geometry Database* supports storing, updating and retrieving the geometry of *BM@N* modules. The developed information system includes the database, intuitive and compact GUI tools and API tools as a set of *ROOT* macros. The experience of the *Geometry Database* design for the CBM experiment has been applied to this development. Improvements in application and GUIs based on *BM@N* users responses are assumed. The *Geometry Database* is being prepared for production.

Acknowledgement

The work was funded by the Russian Foundation for Basic Research (RFBR) grant according to the research project 18-02-40125.

References

- [1] V. Kekelidze, V. Kolesnikov, A. Sorin, EPJ WoC. **171**, 12001 (2018)
- [2] M. Kapishin, Nuclear Physics A **982**, 967–970 (2019)
- [3] K. Gertsenberger, S. Merts, O. Rogachevsky, A. Zinchenko, Eur. Phys. J. A **52**, 16214 (2016)
- [4] M. Al-Turany, D. Bertini, R. Karabowicz, D. Kresan, P. Malzacher, T. Stockmanns, F. Uhlig, J. Phys.: Conf. Ser. **396**, 022001 (2012)
- [5] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov, Physics of Particles and Nuclei Letters **1**, 97–106 (2018)
- [6] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov, EPJ WoC. **214**, 02030 (2019)