

GEOMETRY DATABASE FOR THE CBM EXPERIMENT

**E.P. Akishina¹, E.I. Alexandrov¹, I.N. Alexandrov¹, I.A. Filozova^{1,2,3,a},
V. Friese⁴, V.V. Ivanov^{1,5}**

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

² *Dubna State University, Institute of system analysis and management, 19, Universitetskaya., Dubna,
Moscow region, 141980, Russia*

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

⁴ *GSI, 1 Planckstraße, Darmstadt, 64291, Germany*

⁵ *National Research Nuclear University “MEPhI”, 31, Kashirskoe shosse, Moscow, 115409, Russia*

E-mail: ^a fia@jinr.ru

The last improvements and updates of the Geometry Database (Geometry DB) for the CBM (Compressed Baryonic Matter) experiment are described. The geometry DB is an information system that supports the CBM geometry. The main purpose of the Geometry DB is to provide a storage of the CBM geometry and convenient tools for managing geometry modules assembling various versions of the CBM setup as a combination of geometry modules and additional files. There must be functionality to support different versions of the CBM setup. The ability of setup versions is added. A set of corresponding tools, i.e. a graphical user interface (GUI) and an application programming interface (API), was implemented. Users do not always require regular updates of local CBMRoot from the version control system to solve their tasks (such as simulation or reconstruction). The main goal of the new functionality is to automatically select the setup version when loading the geometry corresponding to the current user environment.

Keywords: CBM, Geometry Database, setup, setup module.

Elena Akishina, Evgeny Aleksandrov, Igor Aleksandrov, Irina Filozova,
Volker Friese, Victor Ivanov

Copyright © 2019 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

The acquisition, storage, processing and analysis of experimental and simulated data are an integral part of a physical experiment. The use of information systems makes a significant contribution to the automation of the physical experiment. Physical parameters and data in experimental studies directly depend on the design of the selected detectors. In high-energy physics there are no major experiments without the use of information systems for various purposes. Such information systems keep the detectors geometry as databases, which store and process all the data about the composition and the structure of detectors. The motivation of this development is described in [1]. The information system CBM Geometry DB solves the following main tasks: storing CBM modules, loading geometry modules for the setup construction, assembling the setup from stored modules and supporting the setup versioning. The development is carried out according to the document “User Requirements Document (URD)” [2].

2. General description of the CBM Geometry DB

This information system is specific and determined by the method of assembling various CBM configuration options. The setup consists of a set of setup modules and additional elements. Additional elements include the detailed information on the distribution of the magnetic field inside and around the dipole magnet and a list of materials that make up the CBM installation.

The issues of developing the requirements and the design of the Geometry DB were presented in detail in the papers [1-4]. The keynote concepts are:

Geometry Module — ROOT file with the detector geometry content.

Setup Module — Geometry module, link to the mother geometry module, its placement in the mother module (transformation matrix).

Setup — Combination of setup modules, representing the full CBM geometry.

The CBM Geometry DB is oriented to three types of users: *Lead Developer*, *Developer* and *CBM user*. A *Lead Developer* is a coordinator who is responsible for the entire Geometry DB and has maximum access rights. Managing the setup (delete any setup, create a new one and approve it) is allowed only for the *Lead Developer*.

A *Developer* is a person responsible for only one of the setup modules. Both the *Developer* and the *Lead Developer* can create, edit or delete the setup module.

A *CBM user* can view the content of the Geometry DB; download the full setup or one setup module that belongs to this setup into the local file system; load the setup into the CBM ROOT framework.

The module “Configure Access” was realized. It manages user accounts, configures administration rights and authorization rules.

According to the Geometry DB functional model [1], the system provides interfaces to view, retrieve and update modules and setups; allows one to store the setup as a combination of setup modules, the magnetic field and materials and save setup modules as ROOT files and the Transformation matrix.

The general object model of the system and the logical Entity-Relationship Diagram were presented in the papers [3] and [4] in detail.

The Geometry DB is implemented as a distributed system. The web server, the database interface, the Database Management System and source data are located on the server host. The local database and software CBMROOT are placed at the user workstation. The CBM user interacts only with the local DB (a replica of the master DB).


The system is implemented as GUI (web interface) and API. The web interface provides the following functions: View, Edit and Download. API is implemented as a macros set of the ROOT environment and provides getting the List of setups, Loading setup. Any macro can be used as an executable file or called from other ROOT macros.

Usually the requirements change during the process of the informational system transfer into the production stage, so the system needs to be refined. Therefore, some changes were made and are described below.

3. Design and implementation updates

The main changes in the Geometry DB are made as a reaction to users' requests, and the most significant one is as follows. Physicists in the CBM collaboration use some software releases in the svn system. There was an agreement that geometry root files would have names that included the name of the whole software system. However, geometry root files are changed more frequently than the whole software release. As a result, geometry root files have the same name, but have different contents of the file and differ only in the svn revision. A user can check out geometry files with some revision value, but cannot not be sure which revision number will be used to maintain consistency of the entire geometry. A user can use the latest geometry files with the same name, but sometimes a user uses some old version of the whole software. In addition, users who do not work on geometry improvements want to use some geometry that can be loaded by default using only the file name. Thus, we need to keep some additional parameter in the database, by means of which we can distinguish between geometries that have the same name (or the tag name in case of a database). We should also implement some rules on how to select the correct instance of the root file by default without using this new parameter.

The **revision** parameter was added in the setup entity to improve the setup flexibility: *Setup(#id, stag, description, author, status, lastmodified, revision)*. According to URD [2], each setup is uniquely identified by a tag value. However, a user have to work with a set of different versions of the setup under the same tag value. All geometry versions related to this tag value should be saved in the database. The decision was to link the setup version with the svn revision number of CBMRoot. Thus, a user is able to load the geometry by tag and revision values. In case of the absence of the revision parameter, the geometry corresponding to the svn revision number of CBMRoot installed on the user workstation is loaded by default. If a user does not use the version control system in his CBMRoot environment, the geometry corresponding to the latest version is downloaded. The web form for the new setup submission allows one to display and input the value for the setup revision (Fig.1).

The Edit Service on the web interface was extended with a tool  'Make a copy of the setup'. This tool creates a new setup on the basis of the existing one. It allows a user to quickly create a setup by editing the existing one. Fig.2 illustrates a fragment of the web interface for creating a new setup *sis100_electron revision 14986* based on the existing one.

The available macros are described in [1]. A new parameter *revision* was added at the macros signature. For example, the macro *bool loadSetup(const char* setupTag, const char* moduleName, long revision)* loads the setup into the CBM ROOT environment by the module *Id*. The geometry can be used in the ROOT framework afterwards. Return FALSE if the setup is not loaded, and TRUE if loading is successful.

The revision value of -1 means getting the revision number from local svn or the last one (if svn does not exist). This functionality allows one to automatically select the setup version during geometry loading (the geometry that corresponds to the current user environment).

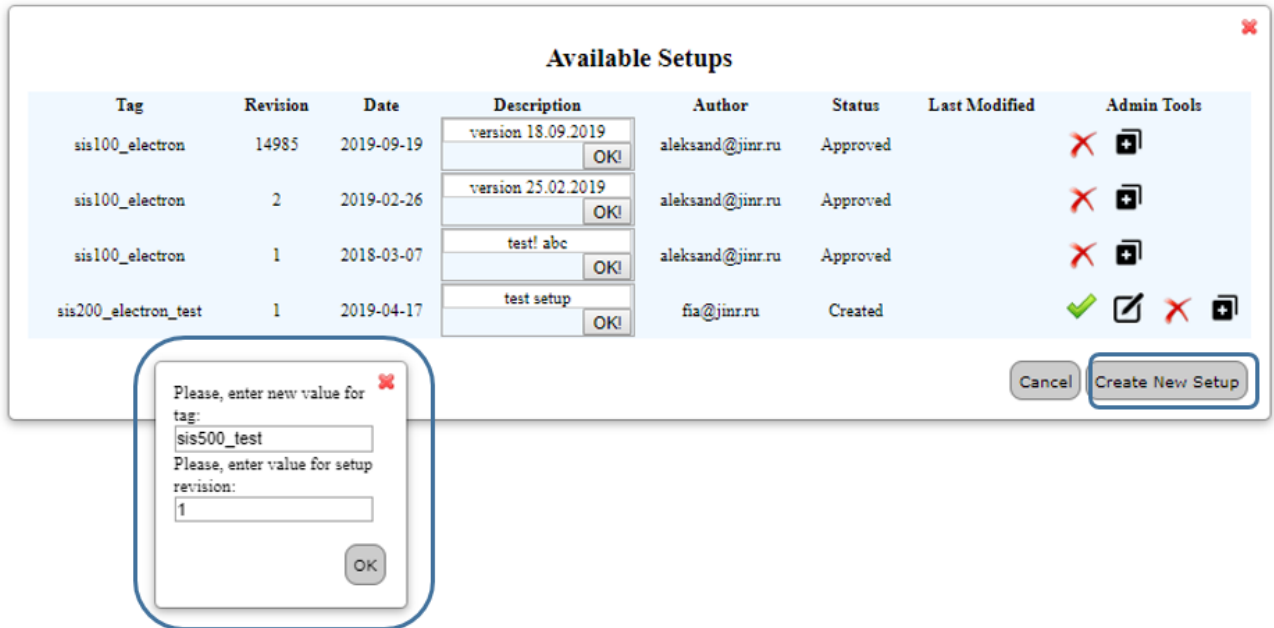


Figure 1. Web interface. Edit Setup

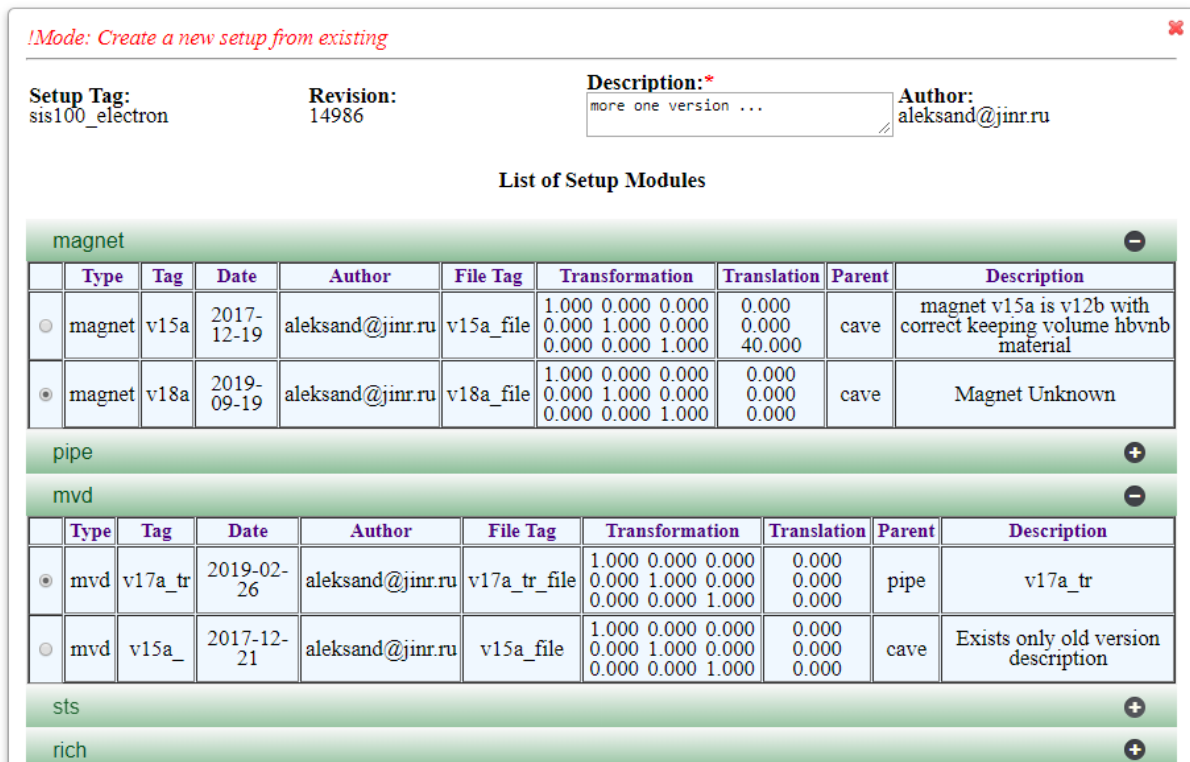


Figure 2. Web interface. Create a new setup from the existing one

The set of macros was expanded with two new ones. A new macro `loadSetupByURL(const char* setupTag, const char* moduleName, long revision, const char* url)` loads the setup into the CBM ROOT framework. The Geometry can be used in the ROOT framework afterwards. The macro `loadSetupByURL` allows a user to load the setup that has the status Created. This functionality is useful for a developer to test the setup before accepting.

The second implemented macro `installLocalDB` installs the sql replica of the full geometry on the user workstation.

4. Conclusion

The Geometry DB was developed for the CBM collaboration, and database improvements based on users' requests are in progress. One of the main changes is the addition of a revision number for the setup object in order to improve the database flexibility and convenience in use. GUI and API were updated accordingly. The svn revision of the whole CBMRoot software was used in order to automate this process. LoadSetupByURL macros were added as well. Using these macros is convenient when creating and testing new setup objects.

References

- [1] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov. Experience of the Development of the Geometry Database for the CBM Experiment//EPJ Web Conf., Volume 214, 2019. - DOI: <https://doi.org/10.1051/epjconf/201921402030>.
- [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] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov. Status of the geometry database for the CBM experiment, CEUR Workshop Proceedings, Vol-2023, pp.144-154 (2017).
- [4] E. Akishina, E. Alexandrov, I. Alexandrov, I. Filozova, V. Friese, V. Ivanov. *Development of the Geometry Database for the CBM Experiment*, PEPAN Letters, **1** (2018)