CBM

User Requirements Document of the Geometry DB for the CBM experiment

Document Version: 0.4

Document ID:

Document Date: 15 February 2016

Document Status: Revised Draft

Abstract

This paper presents the User Requirements Document (URD) of the Geometry DB for the CBM experiment.

Institutes and Authors: Akishina[1] E.P, Alexandrov[1] E.I., Alexandrov[1] I.N., Filozova[1] I.A., Friese[2] V., Ivanov[1] V.V.

[1]JINR

[2]GSI:

.

Table 11 Document Change Record

Title:			
ID:			
Version	**Issue**	**Date**	**Comment**
0	1	17/12/15	Document creation
0	2	19/01/16	Revision V. Friese
0	3	30/01/16	Revision E. Akishina
0	4	15/02/16	Final editorial fixes

1 Introduction

The CBM geometry describes the CBM detector setup on the detail level required for transport simulation (GEANT). It represents the “ideal” geometry in the sense of the construction blueprint. The deviation of the actual geometry from this ideal one (obtained after installation by optical surveillance or alignment procedures) will be stored in a different database (parameter/conditions DB).

The CBM geometry description format ROOT (TGeo classes) is realised as a tree of nodes with mother-daughter relationships. The top level node which comprises the entire geometry is called “Cave”. The second level is made of modules, each standing for a CBM detector or passive system. The delivery and maintenance of the geometry on the module level is in the responsibility of the respective project groups.

1.1 Purpose of the document

This paper presents the User Requirements Document (URD) for the Geometry DB of the CBM experiment. It shall be the basis for the design and implementation of the Geometry DB for the CBM experiment.

1.2 Glossary, acronyms and abbreviations

1.2.1 Glossary

Geometry Module

File in ROOT format with content of detector geometry. The Module is uniquely identified by 3 names:

- Software version (compliance with compiled code);
- Context (e.g., “sis100”, “beamtestCosy14”);
- Running version, reflecting the change of the geometry design during the development phase or modifications (e.g., replacement, exchange of components) in the run phase.

The full name is a concatenation of the 3 names described above.

Each module has a unique module index, realized as a C++ enumerator.

Module Index

Unique identifier of a CBM module.

Setup Module

Geometry module, link to the mother geometry module, its placement in the mother module (transformation matrix or object of class TGeoMatrix).

Setup

Combination of setup modules which represents the full CBM geometry.

Setup Subset

Setup module of a setup identified by setup name and module index.

Role

The Role of a DB user can be one of the following:

- CBM user
- Developer
- Lead Developer

CBM User

A user registered in the Geometry Database. The CBM User can only read data. A CBM User can be a human using a GUI or an application through an API.

Developer

The responsible developer for one of modules.

Lead Developer

Coordinator and responsible person for the entire CBM geometry.

1.2.2 Acronyms and Abbreviations

GSI	Gesellschaft für Schwerionenforschung
DB	Database
CBM	Compressed Baryonic Matter

1.3 References

1

2 General Description

Current situation

1. The geometry of the CBM modules is provided in ROOT files, each containing the top-level volume for the respective module. Each file comes with a running versioning tag, e.g. “sts_v15a.geo.root”.
2. For the transport run the geometry file for each module has to be specified in the run macro. The complete CBM geometry is defined by this set of files.
3. The geometry files are distributed through the software repository. They are not supposed to be changed with repository revisions; versioning happens through the explicit version tag in the file name instead.
4. The user has the possibility to choose a pre-defined setup (e.g., “sis100-electron”) as a set of module geometries. This is realised by ROOT macros to be included from the run macro. They are subject to change with repository revision / software release.
5. The complete geometry (TGeoManager) is constructed from the specified geometry files at the initialisation of the run by the run manager class FairRun through calling the method ConstructRootGeometry() of the registered module objects (class FairModule). It is then stored as parameter container in a parameter file.
6. Later runs (e.g., digitisation, reconstruction, analysis) read the geometry used during transport from the parameter file.

The handling of the geometry files in the repository, so by a versioned file system, is not an ideal situation. It is rather complicated and error prone. Moreover, handling by the software repository does not match the requirement that a given geometry version must not change in time. The distribution of the module geometries through a database thus appears a desirable solution.

2.1 Context of the Geometry DB

The Geometry DB will store the CBM geometry and setup modules. It will also store set-ups as combination of setup modules. The Geometry DB shall provide interfaces to view, retrieve and update modules and setups.

2.2 General capabilities of Geometry DB

The Geometry DB will be accessed by CBM users through a web interface for

viewing and retrieving separate modules and full setups. It will be accessed by batch jobs through a suitable API for download or load any geometric modules or a full setup.

2.3 General assumptions and dependencies

The Geometry DB is used for loading the geometry modules to construct the complete geometry during the initialisation of a CBMROOT run. It gets information about access privileges from the CBM Collaboration Database.

2.4 User characteristics

There are three categories of users of the Geometry DB:

- CBM user
- Developer
- Lead Developer

3 Specific Constraints, Assumptions/Dependencies, Use Cases and Requirements

3.1 Constraints

CO001 Platforms

The Geometry DB should work on all platforms supported by the CBM collaboration. These are currently common Linux flavours (Debian, Ubuntu, Scientific Linux) and Mac OS.

CO002 Performance

The Geometry DB should have high performance, since it should be suitable to be concurrently accessed by a large number of batch jobs.

CO003 Availability

The Geometry DB should have high availability.

CO004 Safety

The Geometry DB must have high fail safety.

CO005 Permission to add modules

Only Developers and Lead Developers can add a new geometry or setup module.

CO006 Add setup

Lead Developers can create a new version of a setup selecting a combination of setup modules.

CO007 Permission to create setup

Only Lead Developers can create a setup.

CO008 Permission to delete

Only Lead Developers can delete a setup.

Lead Developers can delete setup modules.

Developers can delete their setup modules if those setup modules are not belong to any setup.

The permission policy is subject to possible future changes following decisions of the CBM collaboration.

CO009 Network

The Geometry DB should load a setup or a setup subset in the local network area.

3.2 Assumptions and Dependencies

AD001 Interaction with other data bases of CBM

The Geometry DB interacts with the CBM Collaboration Database to detect the role of the user.

AD002 Interaction with other CBM software

The CBM software framework CbmRoot shall use the Geometry DB to load or download a setup or a setup subset.

AD003 Interaction with WEB

The Geometry DB shall be available through the Web for viewing setups, setup and geometry modules.

3.3 Use Cases

UC001 Load Geometry

A CBM User will use the Geometry DB to load a selected CBM setup or setup subset for the use in their own program.

UC002 Download Geometry

A CBM User can download a selected setup or setup subset to the local disk.

UC003 WEB View

A CBM User can view a description of existing setups and setup and geometry modules.

UC004 Add Geometry

A Developer shall create a new version of setup and geometry modules.

UC005 Add Setup Module

Lead developers and Developers shall create a new setup module. They select one of the existing geometry modules and add the name of a setup module and the transformation matrix and select the mother module.

UC006 Add Setup

Lead developers shall create a new setup. They select a set of existing setup modules.

Note Setup modules usually should cover all detectors which means the setup represents a full CBM geometry.

3.4 Functional Requirements

UR001 Setup Access

The Geometry DB shall provide access to load any setup or setup subset stored in the DB.

Priority High

Note Setup subset means the instance of a geometry module which belongs to the corresponding setup module and setup.

UR002 GUI interface

The Geometry DB shall provide a GUI interface to view setups, setup and geometry modules.

Priority High

UR003 API load interface

The Geometry DB shall provide an API (C++) to load any setup or setup subset stored in the DB.

Priority High

Note: ROOT macro (e.g., loading the geometry modules for construction of the complete geometry during the initialisation of a cbmroot run).

UR004 API download interface

The Geometry DB shall provide an API to download a setup or a setup subset into the local computer area.

Priority High

Note: The interface can be used to create a command line xxx to retrieve a given module geometry and store it locally as a ROOT file for inspection.

UR004 Access control

The Geometry DB shall provide access on three levels:

- CBM user: read access;
- Developer: read access; add new geometry; add new setup module
- Lead developer: read access; add new geometry; add new setup module; change or delete existing setup and geometry modules; define, change and delete setups.

Priority High

Note Normally, there will be one person responsible for a given geometry module.

UR005 Multiple access to Geometry DB

The Geometry DB shall allow multiple access for users of any role at the same time. The database should provide a low response time and a high availability.

Priority High

Note Low response time means that this time depends only on the network speed.

UR006 Backup

The Geometry DB shall do regular backups.

Priority High

3.5 Non-Functional Requirements

UR007 Geometry store

The Geometry DB shall store all setups and setup and geometry modules.

.

Priority High

UR008 Consistency of Geometry Module

The Geometry DB shall check if the full name of module is unique and refuse if it is not. The responsibility of Developer is to deliver a correct ROOT file.

Priority High

UR009 Consistency of Setup

The Geometry DB shall assure the consistency of a combination of setup modules.

Priority High

Note The responsibility of the Lead Developer is to create a setup which does not contain any conflicts from the constructional point of view. This means the corresponding setup modules can be correctly loaded into the ROOT environment.

UR010 Consistency of Setup Module

The responsibility of the Developer or Lead Developer is to add a geometry module, its link to the mother geometry module and the corresponding transformation matrix.

Priority High