SpdRoot: the framework for simulation, reconstruction and analysis.

Artur Tkachenko
(avt@jinr.ru)

International Workshop “SPD at NICA-2019”
4 - 8 June 2019

What is SpdRoot ?

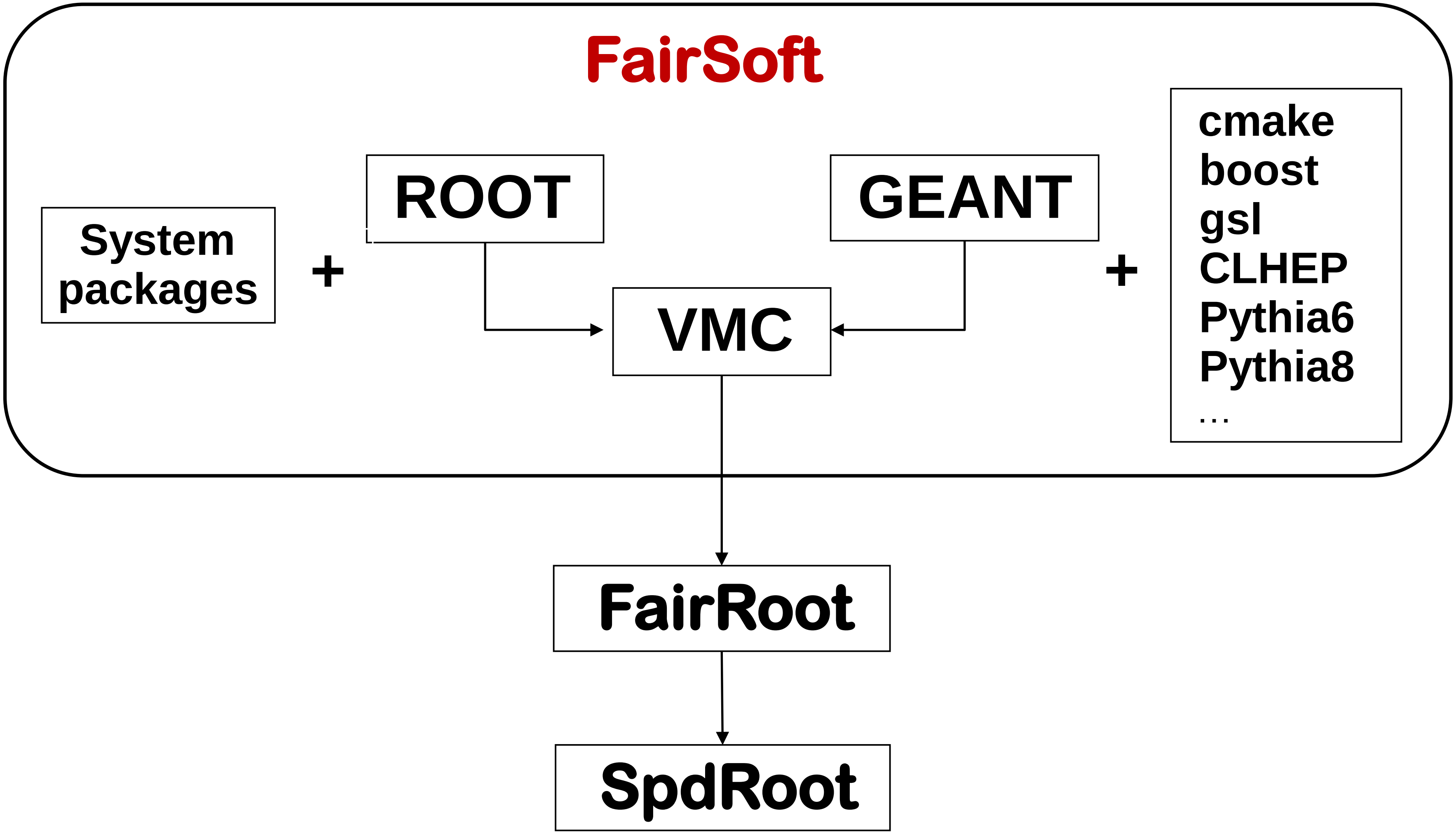
SpdRoot is the FairRoot based project to simulate SPD performances and to evaluate different detector concepts.

The package includes tools for detector simulation, data analysis and reconstruction.

An object oriented code is written in C++ with usage of ROOT and Virtual Monte Carlo (VMC) libraries.

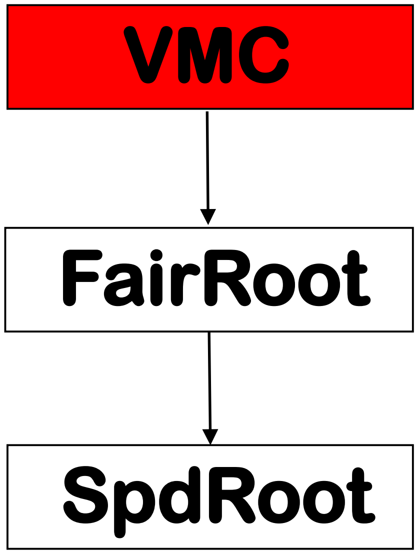
Actual version of the SpdRoot package can be downloaded from:
https://git.jinr.ru/Tkachenko/spdroot

Package dependencies tree



The main FairSoft package objective is to make the installation procedure as easy as possible !

Virtual Monte Carlo (VMC)

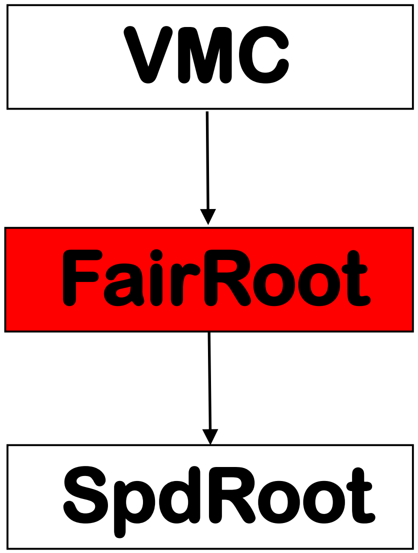


The concept of Virtual Monte Carlo has been developed to allow different simulation programs to separate Monte Carlo transport code (that is simulation of the passage of particles through matter) from user code (such as geometry definition, the detector response collecting, input/output formats etc).

The VMC is based on the ROOT system.

Once the VMC application has been defined, simulations can be run using Root macros (or interactively from the Root UI).

FairRoot



FairRoot has been developed in order to have a common computing structure for the FAIR experiments (PANDA, CBM, R3B).

The FairRoot is an object oriented simulation, reconstruction and data analysis framework based on the ROOT and VMC packages.

The software includes core services for detector simulation and offline analysis of a particle physics data.

Basic idea of FairRoot is to provide a unified package with generic mechanisms to deal with most commonly used tasks in HEP.

SpdRoot: The Points

- Programming code is divided into two parts: pre-compiled code written in C++ and interpreted ROOT- scripts (so-called macros).

- Monte Carlo (Geant4) settings is accessible through the VMC interface.

- The detector geometry is described by ROOT objects using TGeo library (another way to define geometry: gdml/xml or FairRoot-ascii geometry format may be used).

- Simulation parameters and output are saved into the ROOT files (*.root). Data structure is based on well-known ROOT objects like TTree, TBranch, ...

- Simulations can be parameterized, initialized and carried out using Root scripts.

Simulation tasks

Standard questions are:

- What the particle (or list of particles, in general) do you want to transport?
- What is the magnetic field value inside the detector?
- What is the detector geometry?
- What is the output (parameters & data collecting)?

Primary vertex generators (event generators)

The current list of SpdRoot event generators:

SpdPythia6Generator - spd-wrapper for standart Pythia6 generator;
SpdPythia8Generator - spd-wrapper for standart Pythia8 generator (*);
SpdFtfGenerator - spd-wrapper for generator is based on FTF-model;
SpdIsotropicGenerator - isotropic and quasi-isotropic generator;
SpdParticleGenerator - generates one particle or several identical particles;
SpdMultiParticleGenerator - generates a set of any different particles;
SpdEvtBaseGenerator - reads events from the .txt or .root files.

- A global transformation can be applied to the event (to each particle in the list): change vertex position or beam tilt;

- It is possible to use several generators to generate a single event;

- All the parameters are saved in a special object (SpdMCEventHeader) into the output data-file.

Magnetic field

The magnetic field in SpdRoot can be set in one of the following ways:

- **Analitically (by function).** A field at any point is calculated by a previously described function (or algorithm).

- **By a table (map).** The field is get in the nodes of the cubic grid previously recorded in a file.

- **Fields set (multifield).** An arbitrary combination of the two types of fields described above (as well as the corresponding geometric regions).

There are three possible choices for the field region:

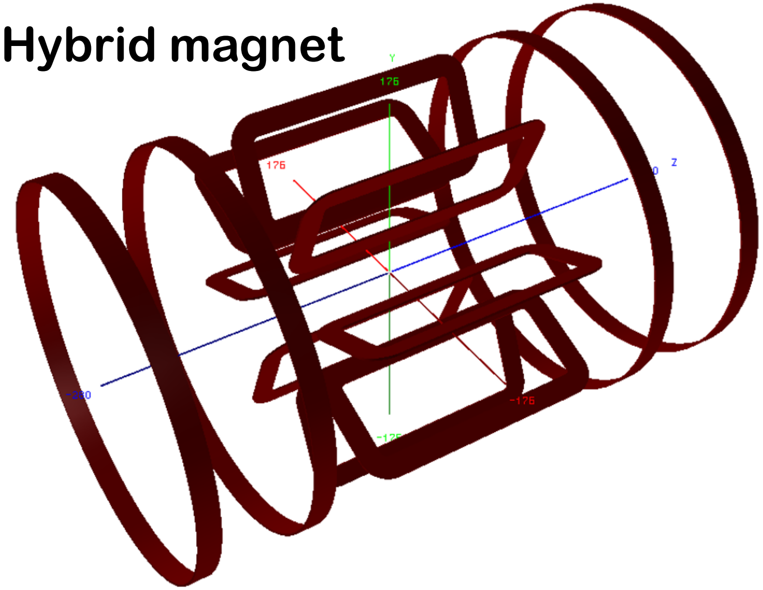
Box - A simple parallelepiped.

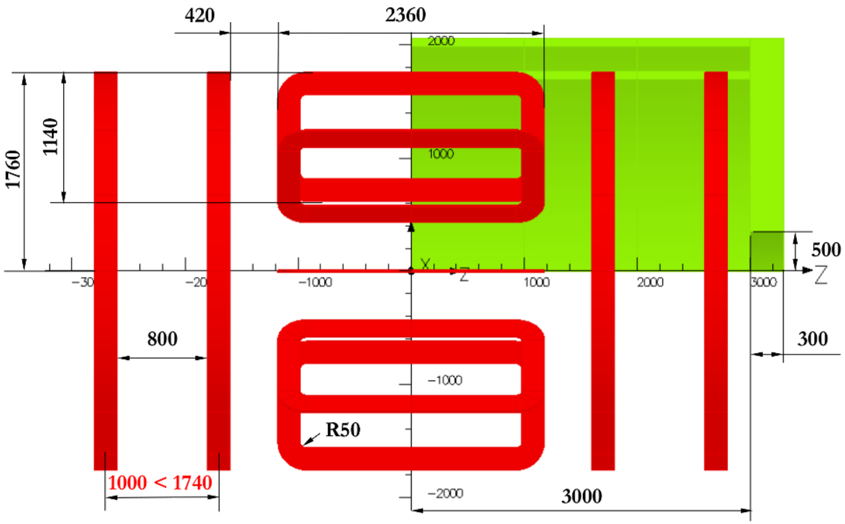
Tube - A cylinder, the axis of which coincides with the coordinate axis Z.

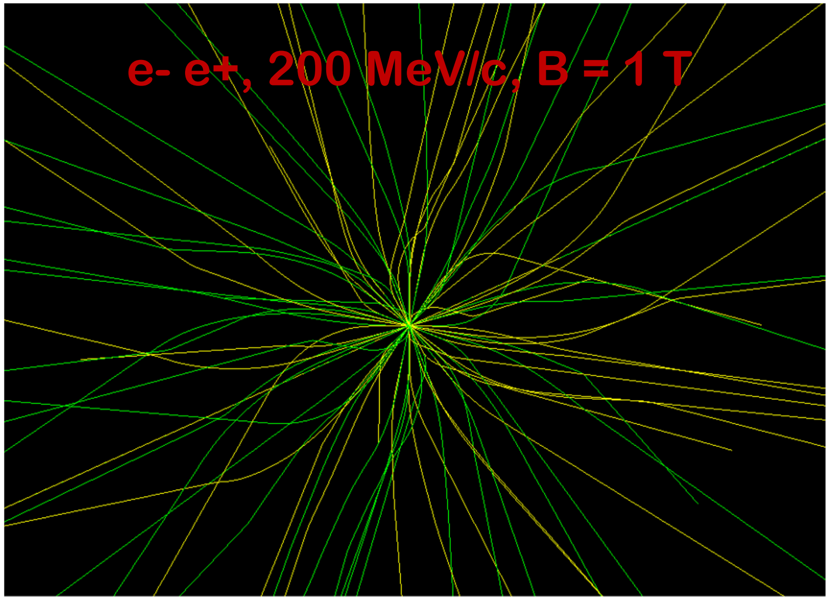
Physical - the region is defined either by the specified name (more exactly, by volume geometry string id – “geopath”) or physical volume, or by material.

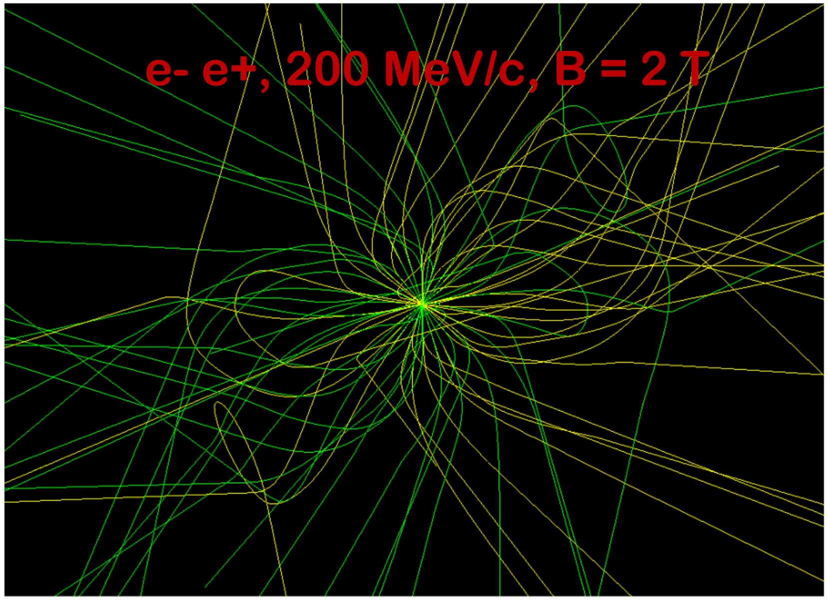
Magnetic field

Hybrid magnet

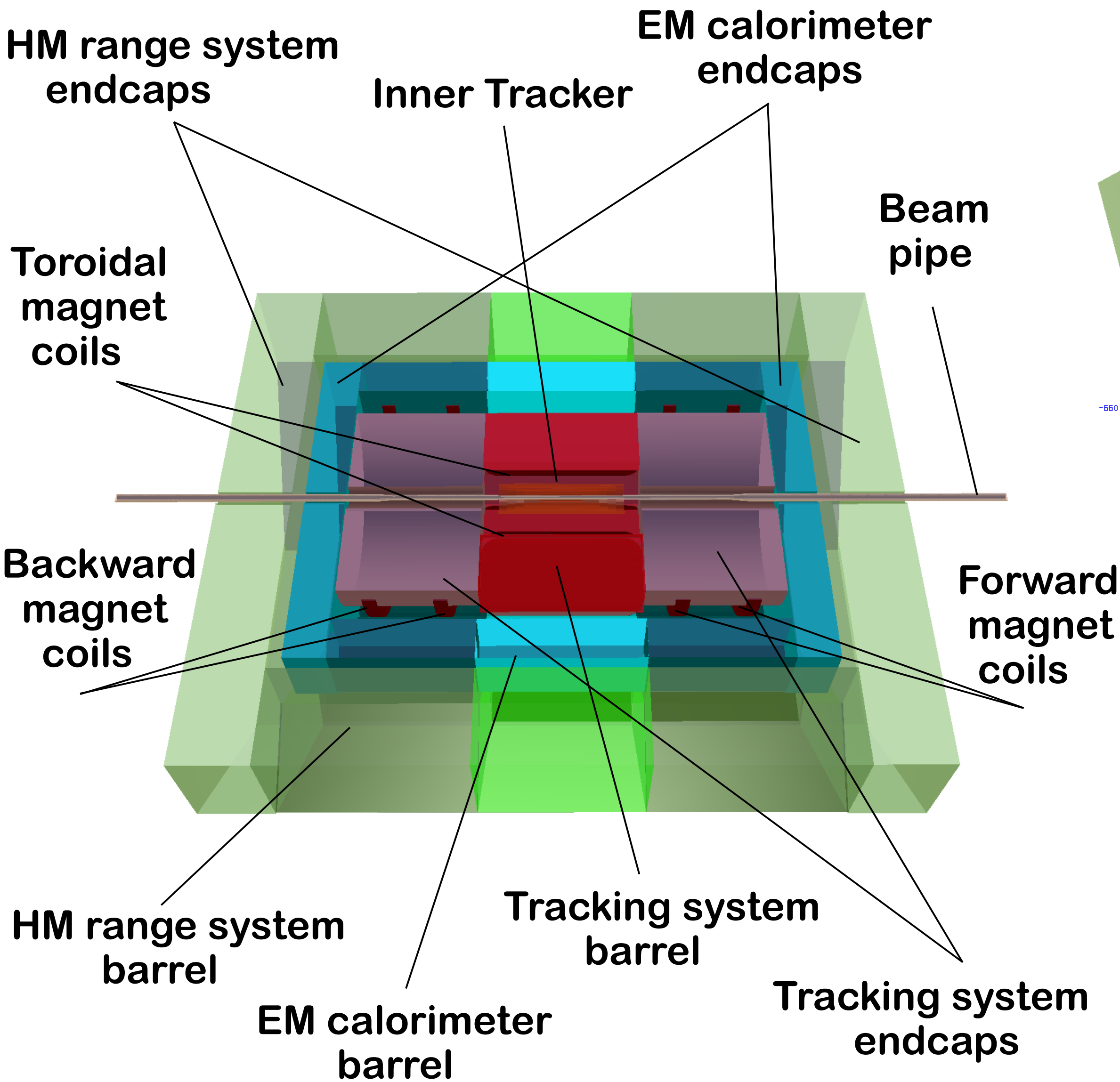


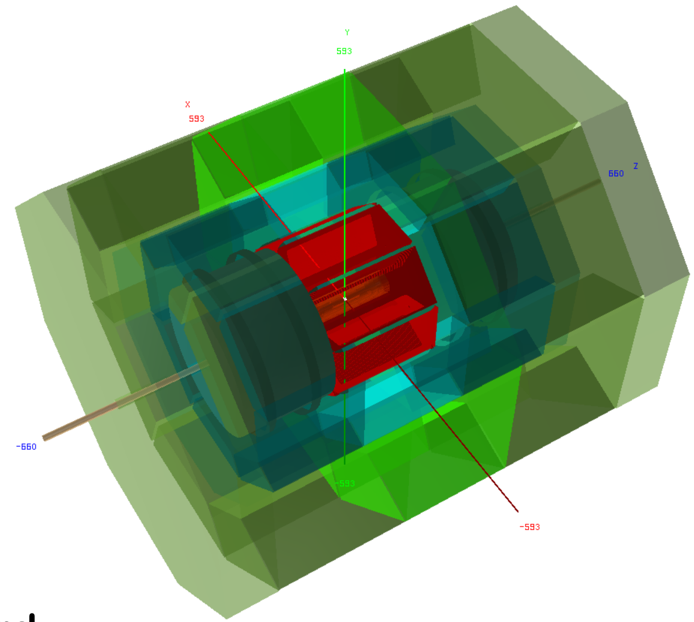


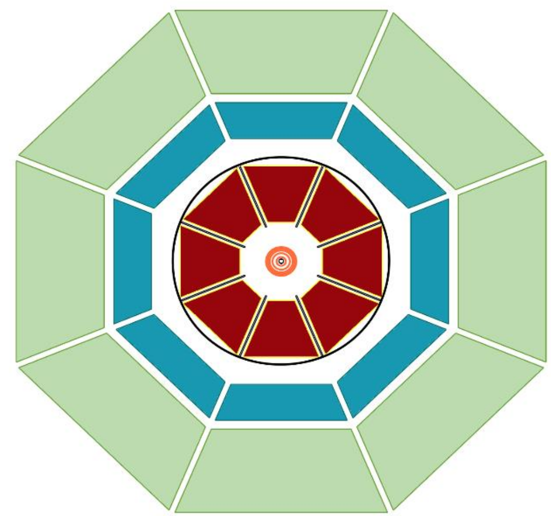




Detector geometry







Detector geometry : features

Modularity:

The setup configuration is determined by a set of modules.

In the narrow sense, the module is simply program code containing a description of the geometry of a setup part that can be physically and functionally considered as a whole.

An examples of modules: pipe, magnet (so-called “passive” modules, because of there are no detective elements inside), vertex detector, tracking system, electromagnetic calorimeter, etc (“active” modules).

In a broad sense, the module is a part of program code that, in addition to describing the geometry, also provide the collection and some primary processing of useful physical information obtained during the simulation in the format suitable for saving to a file.

Any module can be easily removed from the process (or replaced with another) without destruction of overall activity anytime!

The module also forms a list of its parameters, which will be added to the output file.

Detector geometry : features

Parametrization:

One of the most important properties of SpdRoot is the ability to vary the geometry of the detector.

At the moment, the design of the detector is still discussed and many setup parameters should be optimized taking into account the physical program of the experiment.

The ROOT package contains tools for creating geometry of the most common type and, as a result, requires a lot of parameters for element-by-element description of the geometry structure. Nevertheless, the number of parameters can be significantly reduced by using parameterization not for the direct description of all elements of the geometry, but only for special algorithms with the help of which this geometry can be created.

With the successful choice of parameterization, it is possible not only to reduce the total number of parameters, but also to make the design description more natural and easy to understand. Further, this can significantly help in data analysis.

Detector geometry : features

SpdRoot has created and applied a number of tools for creating parametrizations, which allow varying the geometry of the detector in very wide range.

Before starting the simulation, the setup geometry parameters can be changed easily. By the end of the simulation the entire list of parameters will be saved to the output file and can be used to restore the geometry when analyzing the data.

Output: files

Standard simulation output consists of root-files of two type:

Data

- run.root
  - cbmroot [main tree]
    - SpdMCTrack [list of particles]
    - TrackerPoint [TS endcaps]
    - SpdTsTBPoint [TS barrel]
    - SpdEcalTBPoint [ECAL barrel]
    - SpdEcalTECPoint [ECAL endcaps]
    - SpdRsTBPoint [RS barrel]
    - SpdRsTECPoint [RS endcaps]
    - SpdItsPoint [inner tracker]
    - MCEventHeader [event info]
    - GeoTracks [TGeoTracks]

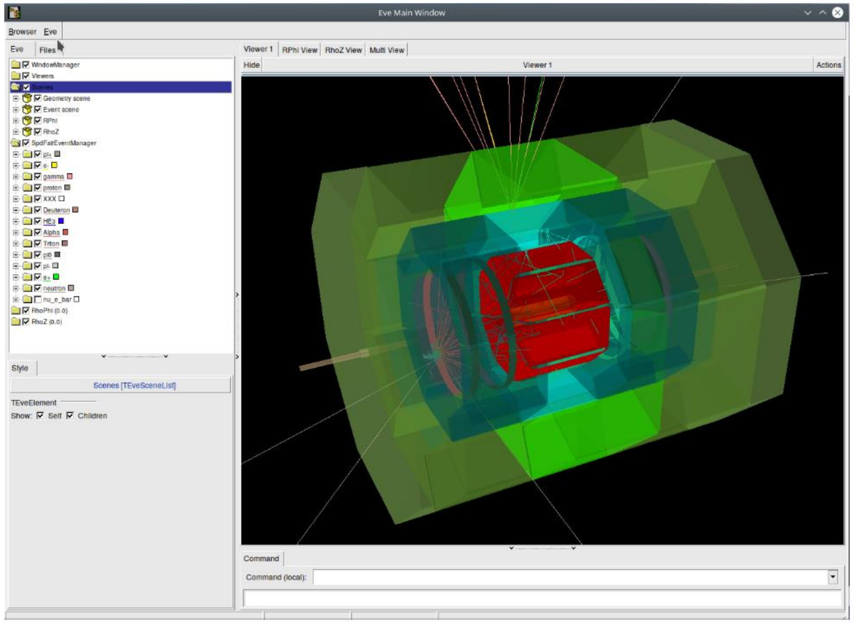
Parameters

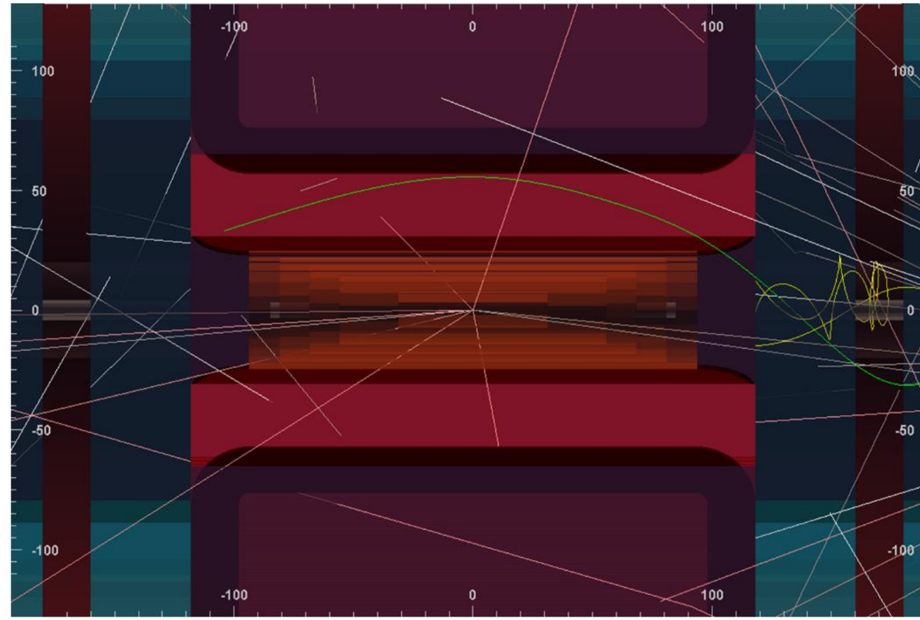
- params.root
  - FairBaseParSet [list of modules]
  - FairGeoParSet [TGeoManager]
  - SpdFieldPar [magnetic field]
  - TrackerParSet [TS encaps]
  - TsTBParSet [TS barrel]
  - EcalTBParSet [ECAL barrel]
  - RsTBParSet [RS barrel]
  - EcalTECParSet [ECAL endcaps]
  - RsTECParSet [RS endcaps]
  - ItsParSet [inner tracker]
  - PrimGenParSet [event generators]

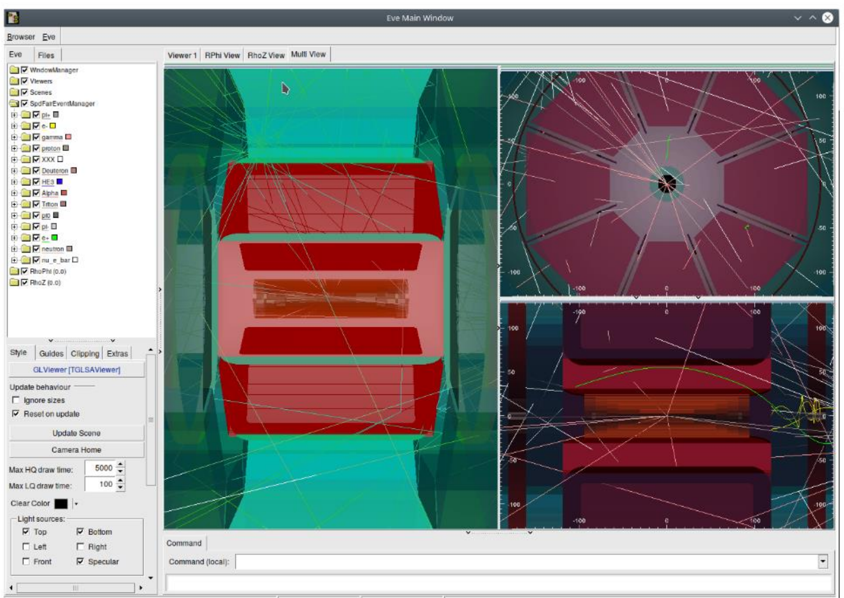
Fast check for the data-file content:
macro/analysis/CheckOutputData.C

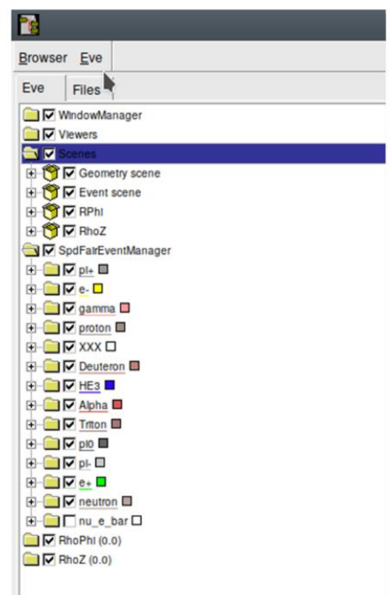
Fast check for the file with parameters:
macro/analysis/CheckOutputParams.C

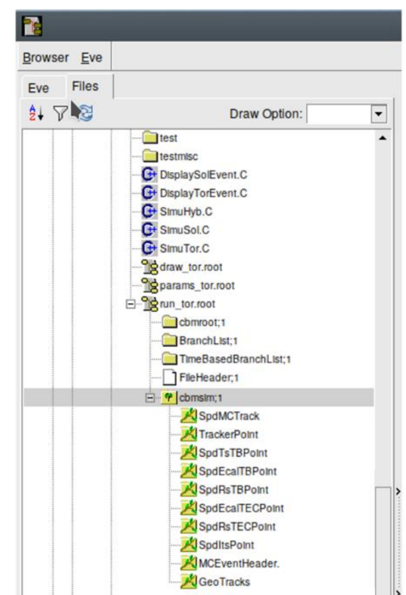
Output: Event viewer











Event reconstruction and analysis (in progress)

Goals to be achieved:

- Restore/check simulation settings (geometry, generators, field, MC-parameters, etc.);
- hit making;
- clustering (for electromagnetic calorimeter);
- track finding/matching;
- track parameters reconstruction;
- particle identification;
- …

ROOT scripts (macros)

SpdRoot interpreted scripts are placed in the directory

spdroot/macro

- **primgen/** : build and test primary generators;
- **field/** : build and test the magnetic field;
- **geom/** : construct and check the detector geometry for general detector and for its parts separately;
- **test/** : simulation and tests for some auxiliary detectors;
- **run/** : additional scripts for simulations;
- **analysis/** : scripts for parameters and data checking, reconstruction and analysis.

Run-script with the actual version of the detector geometry is macro/SimuHyb.C (for more details see SpdRoot manual, chapter "How to run").

Documentation

Information you may find inside the package:

The project manual including the installation guide and "how to“ (in english and russian):
spdroot/doc/SPD_en.pdf
spdroot/doc/SPD_ru.pdf

A brief description for the current update (what's new?):
spdroot/ReleaseNotes.txt

The detailed descriptions for the most important and massive updates (in english and russian):
spdroot/doc/RelaseNotes/ReleaseNotes.<version>_en.pdf
spdroot/doc/RelaseNotes/ReleaseNotes.<version>_ru.pdf

Also, it is possible to create a list of SpdRoot classes in the html format using Doxygen (see SpdRoot manual how to make this).

Thank you for your attention!