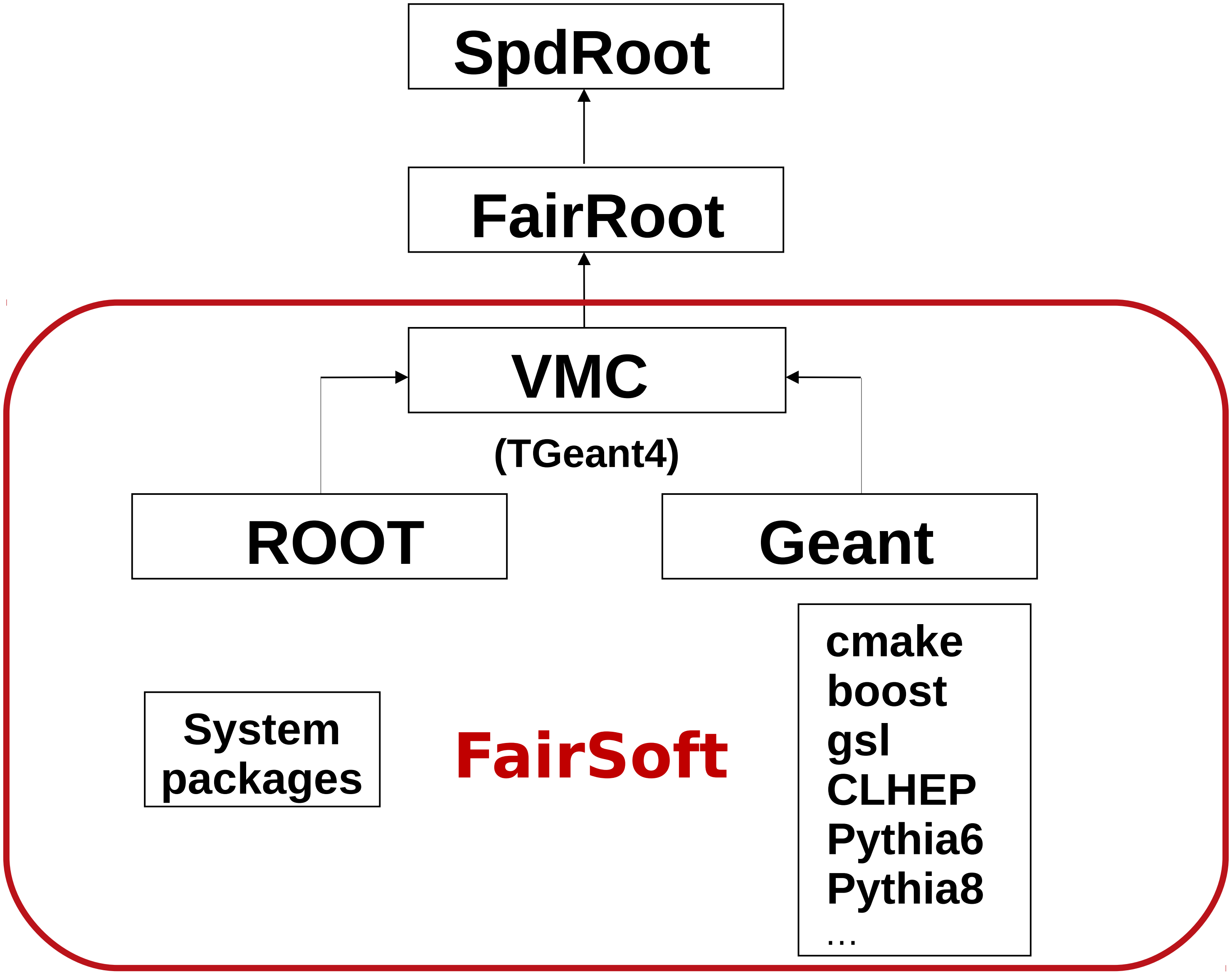
SpdRoot: the framework for simulation, reconstruction and analysis.

Artur Tkachenko

SPD Collaboration Meeting
8 - 10 June 2021

SPD project offline software



SpdRoot

FairRoot

VMC

(TGeant4)

ROOT

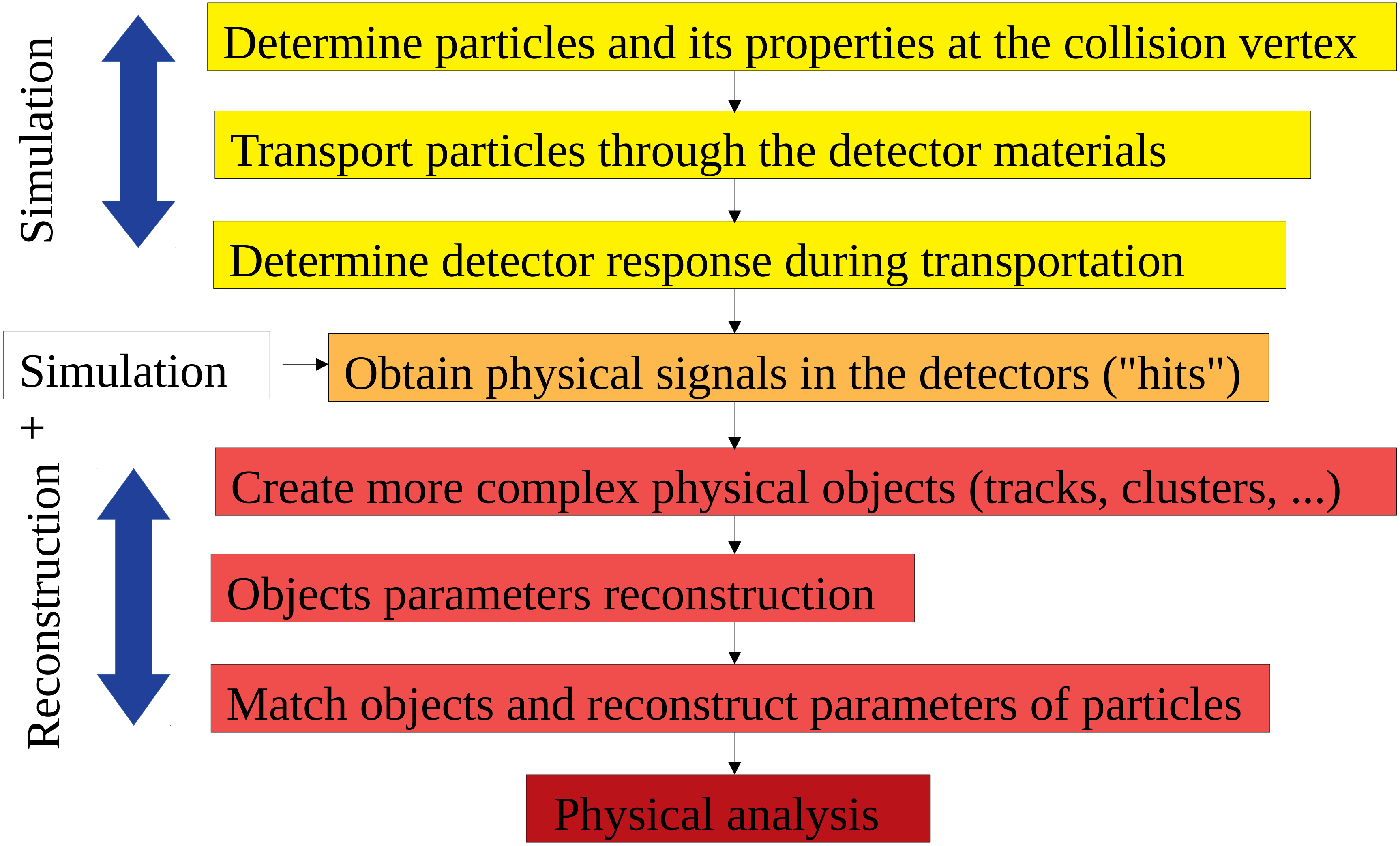
Geant

System packages

FairSoft

cmake
boost
gsl
CLHEP
Pythia6
Pythia8
…

Common view to the data simulation, reconstruction and analysis process



Simulation

Determine particles and its properties at the collision vertex

Transport particles through the detector materials

Determine detector response during transportation

Simulation

Obtain physical signals in the detectors ("hits")

+

Reconstruction

Create more complex physical objects (tracks, clusters, ...)

Objects parameters reconstruction

Match objects and reconstruct parameters of particles

Physical analysis

Simulations (I)

What kind of initialization for proper simulation is required?

* **The package for particle transportation** through the SPD setup: ***Geant4***.

* **Setup geometry.**

- Detectors: VD, STS, ECal, RS, (TOF), (ZDC), (BBC), etc.
-"Passive" modules: magnet, beam pipe and elements of design.

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

* **Magnetic field.**

- Configuration: field map, constant field, …
- Options: field scaling, field region, ...

* **Event generator** (generate particles in collision vertex).

- Physical generators (Pythia6, Pythia8, FTF, UrQMD);
- Generators for precise investigations (isotropic, particle, ascii)

We can use more than one generator at the same time.

* **External decayer** for a special cases should be defined (J/ψ decay)

* **Physics**: the standard Geant4 settings (physics list selection, cuts applying, etc.)

Simulations (II)

Output consists of files of two types:

1) **Data file.**

* ***ROOT TTree "spdsim"***:
  - data branches with raw detector responce produced by Geant4 («MC-points»; separate data branch for an each SPD detector);
  - branch with full list of produced particles («MC-tracks»)
  - information about primary vertex (given by event generator).
* some auxiliary information about ROOT folder structure.

2) **Run parameters.**

- full list of all detectors, "passives" and its parameters (first of all, geometry settings);
- actual geometry that was used during the simulation process (via TGeoManager object);
- primary generator parameters.
- magnitic field type and parameters.

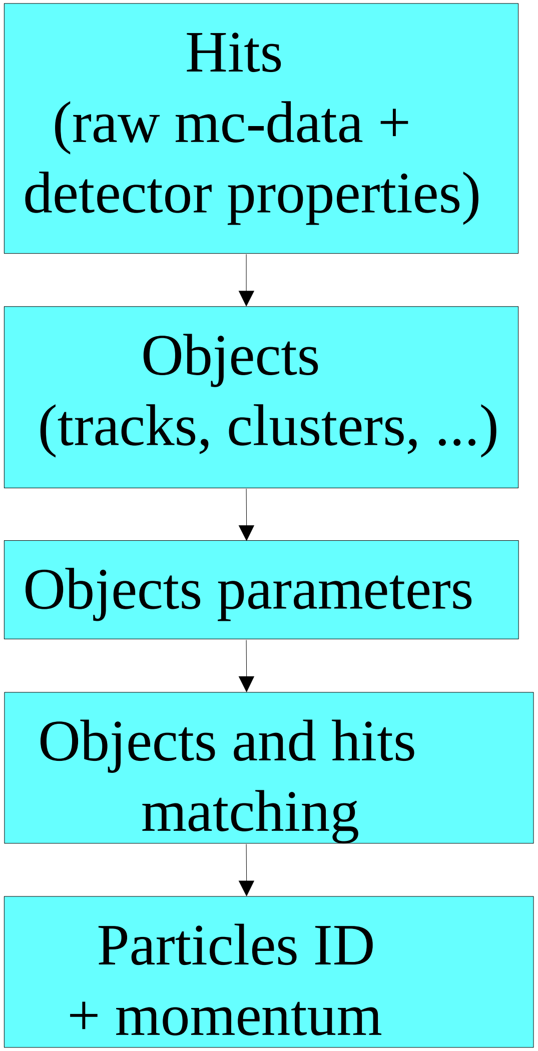
There are a lot of examples for simulation, unit testing and viewing in the SpdRoot package!

Reconstruction (I)

General remarks:

- Reconstruction process is presented as a list of individual subtasks that are performed in a certain order.
- All tasks have to be defined and initialized in an executable ROOT macro.
- Each task has its own set of options and settings.
- As a rule, task completion result is data object written to a separate branch in the output ROOT file tree (TTree).
- The input data for each task may be either the data obtained by simulations or the reconstructed one.
- A single branch with data in the output file should be the result of performing multiple tasks.
- The reconstruction process can be interrupted and then continued with any task.

Reconstruction chain (data flow):



Hits
(raw mc-data +
detector properties)

Objects
(tracks, clusters, ...)

Objects parameters

Objects and hits
matching

Particles ID
+ momentum

Reconstruction (II)

Tasks:

0) **IntermediateStage.Hits**

Tasks: hits producing, mc-truth objects creation.
Algoritms: «raw mc-data»-to-«mc-hits» converters.
Input: raw mc-data & realistic information about detectors (resolution).
Output: - mc-hits for VD, STS, ECal, RS;
- mc-truth: Event, Particles, Vertices.

1) **Reconstruction.Tracking**

Tasks: track finding (pattern recognition) + track fitting.
Algorithms. track finding: mc; track fitting: ***GenFit2*** *package*.
Input: mc-hits for VD + STS.
Ouput: Tracks + parameters.

2) **Reconstruction.Vertices**

Tasks: Primary and secondary vertices reconstruction.
Algorithms: PV: track approximation; SV: ***KFParticle*** *package*.
Input: Reconstructed tracks.
Ouput: Primary and secondary vertices + parameters.

Reconstruction (III)

Tasks:

3) **Reconstruction.ECal**

Tasks: clusters finding, particle reconstruction: real + mc-truth.
Algorithms: clustererization, define particle parameters (real + mc-truth).
Input: mc-hits for ecal.
Ouput: clusters, reconstructed cluster parameters (particles + mc-truth).

4) **Reconstruction.RS**

Tasks: cluster finding, particle reconstruction: mc-truth.
Algorithms: pseudo-algoritms for clustererization and particle identification.
Input: rc-hits for muon/hadron range system.
Output: clusters, reconstructed cluster parameters (particles + mc-truth).

5) **Reconstruction.PID:** *unfortunally, nothing has been done yet :(*

6) **Reconstruction.Matching&PID:** pseudo-algorithms only (mc-truth)

All the reconstruction algorithms (except "pseudo-", of course) are being developed taking into account further working with real experimental data

Analysis

* Useful tools:

- EventHelper: some useful functions and print methods for event data objects.
- DataReader: simple tool for data viewing.
- DataIterator: tool for comfortable data accessing "event by event" from user's program (also it may be used for data merging and event selection).
- TrackPropagator: tool for charged particle transport in the magnetic field.
- ...

* "Full chain" examples (simulation, reconstructruction and physical analsys):

spdroot/macro/examples/

- /chic χc -> γ J/ψ -> γ μ+ μ- events
- /jpsi-ee J/ψ -> e+ e- events
- /jpsi-mumu J/ψ -> μ+ μ- events
- /ecal searching for η-meson decay products in the ECal
- /K0decay searching for secondary vertices

Development infrastructure

* **Git repository**: http://git.jinr.ru/nica/spdroot

* **Documentation (Wiki)**: https://git.jinr.ru/nica/spdroot/-/wikis/home

* **Pre-compiled software**: CVMFS (Ubuntu, CentOS7, SL6)
+ Docker containers

* **Release policy:**
- production branch (master) - major improvements following the release plan or important bugfixes;
- development branch - current development under continuous testing;
- individual developments in separate branches or forked repositories.

* **CI**: Fullchain simu+reco test at every merge request.

* **Current release** (master) - 4.1.0 of 30 March 2021.

Summary

* SpdRoot is currently the main framework for the SPD full simulation, reconstruction and data analysis.

* SpdRoot provides the necessary tools for physics performance study and the detector optimization

* SpdRoot is being constantly developed and improved, but the manpower is very scarce

Most urgent tasks at the moment:

- geometry updating,
- adding TOF detector within the simulation chain,
- further development for the detector response simulation and hit production,
- PID algorithms (dE/dx and TOF),
- track finding algorithm,
- realistic processing for muon/hadron range system data,
- reconstructed object matching

We cordially invite you to take part in this work!