SpdRoot for Physics Analysis

Igor Denisenko
iden@jinr.ru

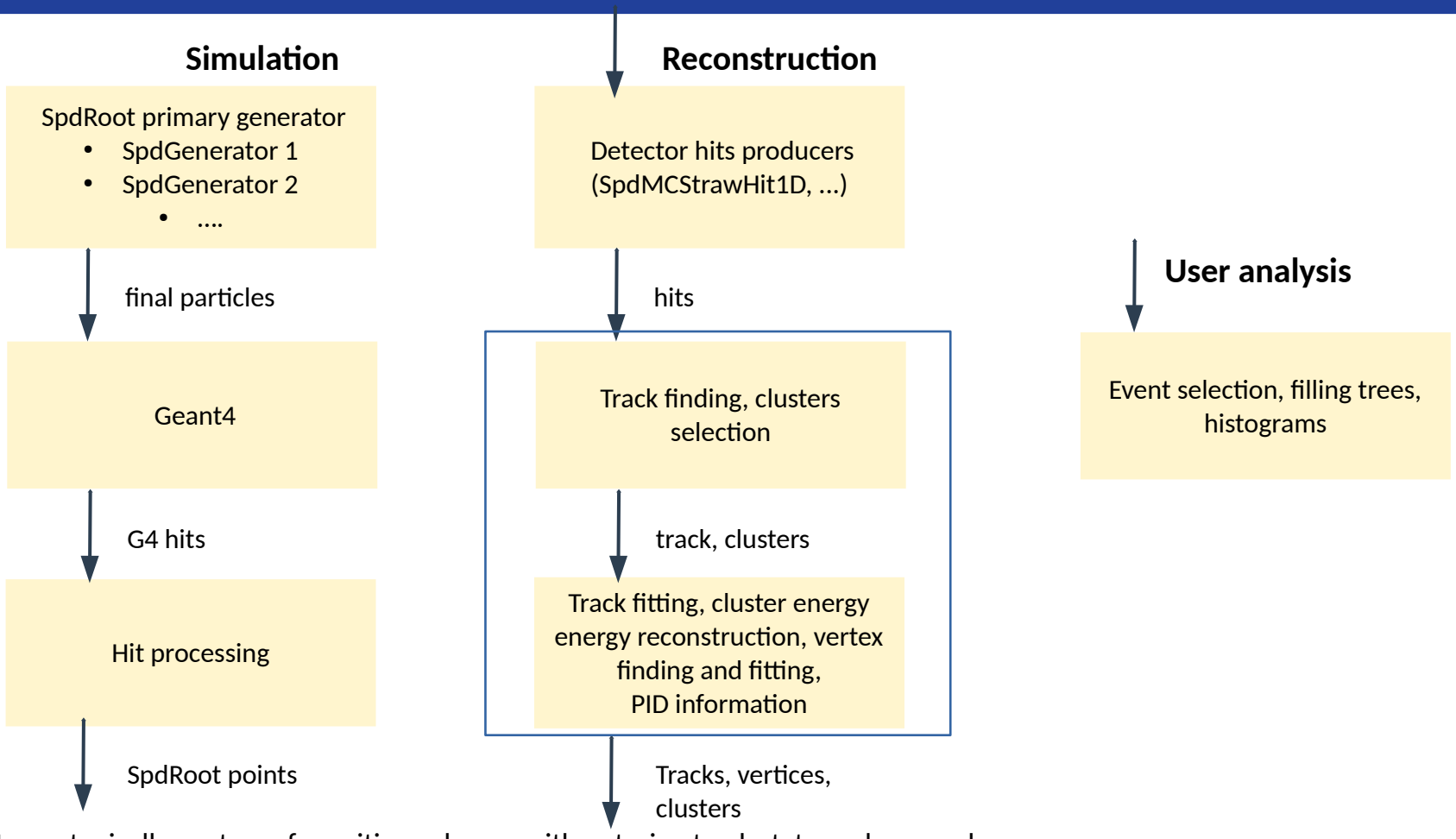
SpdRoot Workshop

16.04.2025

What is SpdRoot?

- SpdRoot is our only physics simulation and analysis tool at least until the end of this year. It's by far not an ideal piece of software, we will try to avoid its issues in SAMPO
- SpdRoot is a collection of c++ classes (mostly accessible from ROOT interpreter) and external packages (GenFit2, KFParticle, …) on top of the FairRoot framework
- SpdRoot now provides
  - event generation with Pythia6/8 and FTF
  - detector geometry with materials and magnetic field map
  - particle propagation through detector with Geant4
  - building detector hits based on the Geant4 hits
  - parts of event reconstruction machinery (particular cases of track finding, track fitting, vertex finding and fitting, clustering and energy reconstruction in Ecal…)
- spdroot.py runs rootlogon.C, "user script.C:, rootlogoff.C
- One can't avoid exploring SpdRoot source code in his/her work
- SpdRoot code bases contains a lot of legacy code

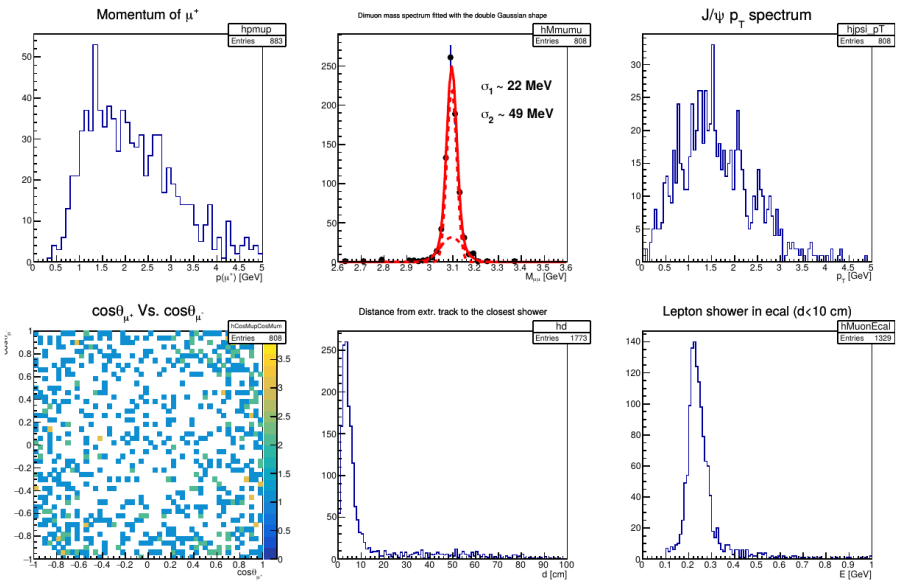
SpdRoot simulation, reconstruction, and analysis chain (schematic view)



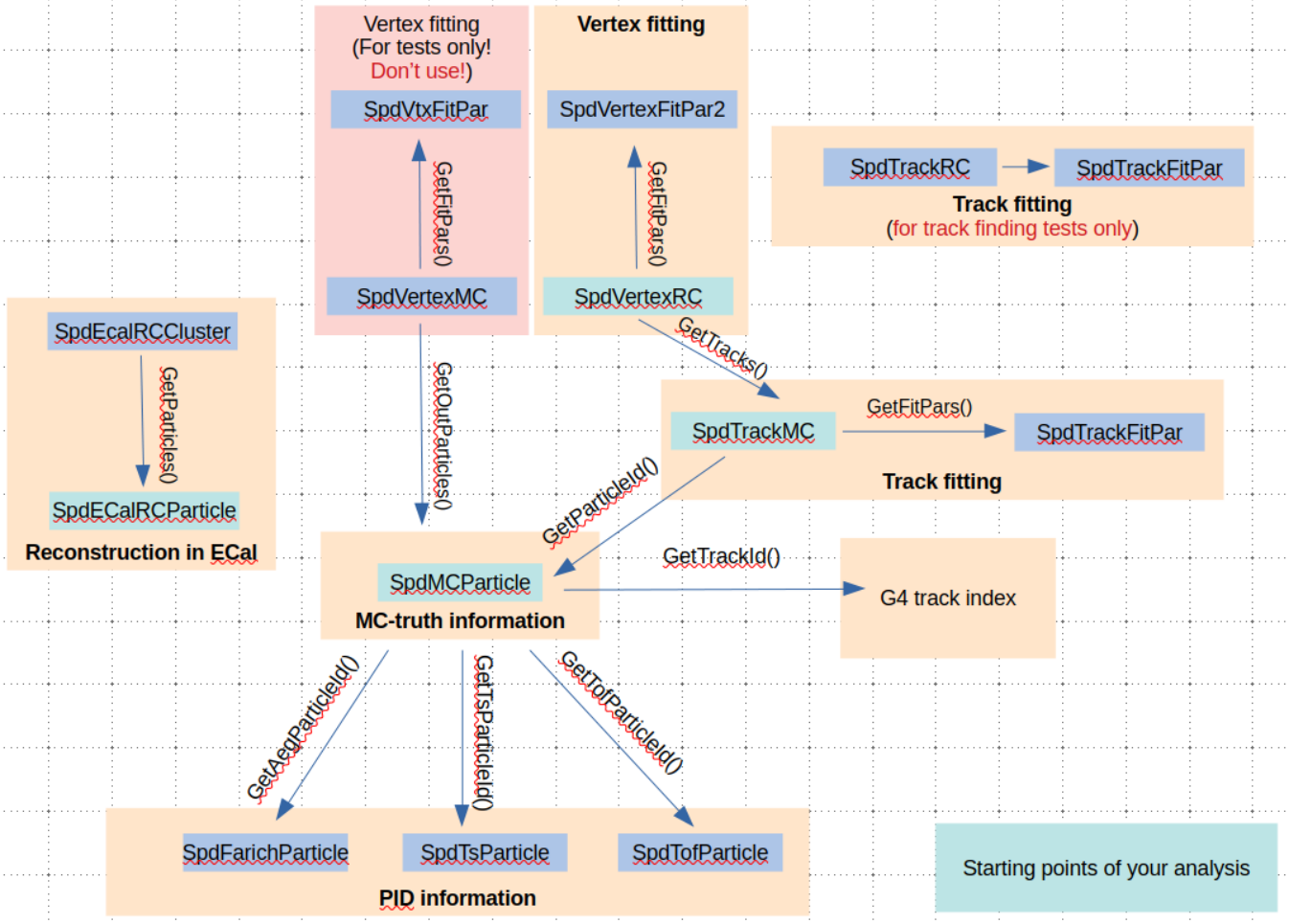
SpdRoot points are typically centers of sensitive volumes with entering track state and energy loses

macro/examples/jpsi-mumu

- **simu.C**: seed → (simu.root, params.root), params.root stores geometry options, magnetic field, etc…
  - magnetic field map description
  - geometry and material description
  - decayer configuration
  - event generator configuration
- **reco.C**: (simu.root, params.root) → reco.root
  - a list of tasks (FairTask derived) that produce output objects (usually TClonesArray collections) based on input from simu.root
- **analyze_jpsi.C**: (reco.root. params.root) → histograms
- **run.sh** runs all this scripts in sequence



Event model for physics analysis



SpdMCDataIterator

- Reconstruction results are stored in TClonesArray collections in the reco.root
- **SpdMCDataIterator** iterates over events and helps to read root branches from the reco-file (e.g. const TClonesArray* SpdMCDataIterator::GetTracks())
- See examples in macro/examples in the SpdRoot source directory

```
DATA BRANCH NAME:     BRANCH OBJECT TYPE:

"MCEventHeader."      FairMCEventHeader
"MCEvent."            SpdMCEvent
"MCParticles"         TClonesArray[SpdMCParticle]
"MCVertices"          TClonesArray[SpdVertexMC(+SpdVertexFitPar)]
"ItsMCHits"           TClonesArray[SpdMCSiliconHit]
"TsMCHits"            TClonesArray[SpdMCStrawHit(1D/2D)]
"EcalMCHits"          TClonesArray[SpdEcalMCHit]
"TofMCHits"           TClonesArray[SpdTofMCHit]
"RsMCHits"            TClonesArray[SpdRsMCHit]
"BbcMCHits"           TClonesArray[SpdBbcMCHit]
"AegMCHits"           TClonesArray[SpdAegMCHit]
"ZdcMCHits"           TClonesArray[SpdZdcMCHit]
"MCTracks"            TClonesArray[SpdTrackMC(+SpdTrackFitPar)]
"RCVertices"          TClonesArray[SpdVertexRC(+SpdVertexFitPar)]
"RCEcalClusters"      TClonesArray[SpdEcalRCCluster]
"RCEcalParticles"     TClonesArray[SpdEcalRCParticle]
"MCEcalClustersInfo"  TClonesArray[SpdEcalClusterMCInfo]
"MCEcalParticles"     TClonesArray[SpdEcalMCParticle]
"MCRsClusters"        TClonesArray[SpdRsMCCluster]
"MCRsParticles"       TClonesArray[SpdRsMCParticle]
"TsParticles"         TClonesArray[SpdTsParticle]
"TofParticles"        TclonesArray[SpdTofParticle]

See reco/tools/SpdMCDataIterator.h for details
```

Important notes

- **SpdTrackMC** provides track fit results (see SpdTrackMC.h) as
  - InitialState (initial guess before KF)
  - FisrtState (state in the first measured point)
  - LastState (state in the last measured point)
  - FinalState (state in the primary vertex if it's found)
- You can use **SpdTrackPropagatorGF** to propagate track state to space point, line or plane (e.g. useful when not primary vertex is found, see macro/performance-tests for track fitting)
- One can access **PID** info via likelihoods from
  - SpdTsParticle
  - SpdTofParticle
  - SpdFarichParticle
- The default secondary vertex reconstruction is based on MC-truth PID. One can predict the background types or use KFParticle in the user analysis part (see find DecayV0.C, recommended)
- **Event generator** and **decayer**
  - event generator describes particle production in the small vicinity of the primary vertex
  - decayer decays long living particles propagated by Geant4 (D, KS, Lambda, …) and needs to be configured separately (see wiki)
- Location of **up to date simulation** and **reconstruction scripts** can be found at the production page (https://docs.google.com/spreadsheets/d/1JWob53dfwMvTlmdsGncwQmPeVbmzFKuD8DHE4rYsJFw/edit?gid=0#gid=0).

People you may address for specific details

- MAPS/DSSD (geometry, hit production) – Artem Vasyukov
- TS (geometry, hit production, dE/dx PID) – Ruslan Akhunzyanov
- TOF (geometry + simplified PID) – Artem Ivanov
- ECal (geometry + reconstruction) – Andrey Matsev
- RS (geometry + clustering (in progress)) – Alexander Verkheev
- FARICH (geometry, hit reconstruction, PID) – Artem Ivanov
- BBC (geometry + hit production) – Arkadiy Terekhin
- Track fitting, status of track finding, primary vertex finding and fitting – Vladimir Andreev
- Secondary vertex – Vladimir Andreev, Amaresh Datta, Natalia Rogacheva, Daniil Gubachev
- Status of muon identification – Ivan Yeletskikh, Alexander Verkheev, myself

You may search their talks with google (e.g. “Andreev site:indico.jinr.u”) or contact them directly

Other notes

- A lot of information can be found at the SpdRoot wiki page (JINR account not required): https://git.jinr.ru/nica/spdroot/-/wikis/home
- Please get SSO and contribute :)
- **Experiment** with modifications of the SpdRoot source code. Adding “prints” (most SpdRoot objects have a useful Print() function) and “getc” is super powerful tool
- Once you properly configured **VSCode**, you have several super useful shortcuts for code navigation:
  - Ctrl+P for fuzzy search
  - F12 to jump to the object/method/class definition
  - Alt+o to switch between the .h and .cxx files
- There is a forum (not active, SSO required) and general chat on simulation in Telegram (join via https://t.me/+eyxg3d65vhdjMjQy)