Possible improvements of the SPD offline software

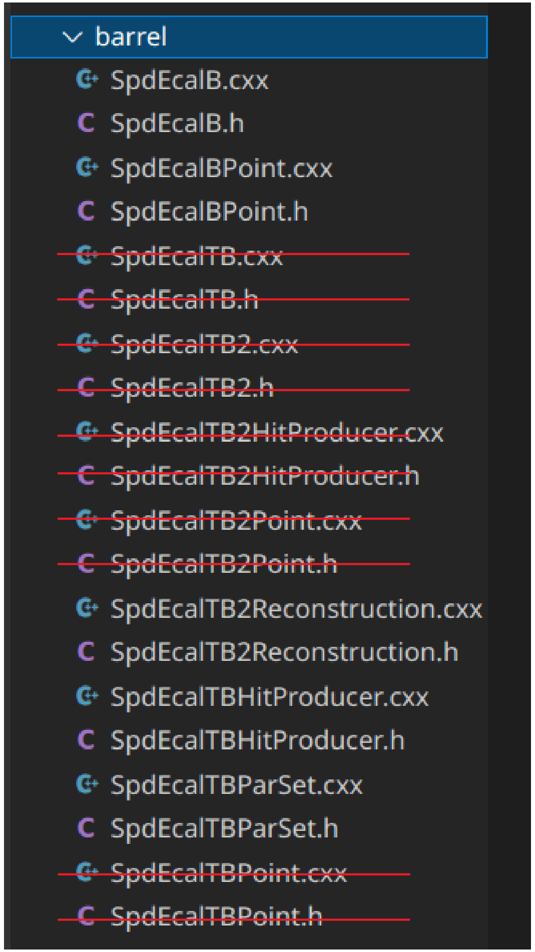
Igor Denisenko
iden@jinr.ru

SPD Software and Computing Meeting
17.05.2022

Motivation

- The software has been developing to get numbers needed for CDR/TDR ASAP.
- For the moment we have a number of points to improve, some points should be addressed not to save huge amount of time in future:
  - Removal of the legacy code (we have a lot!). Part of code is working but not documented = will never be used.
  - Naming conventions.
  - Complicated geometry description.
  - Design mistakes.
  - No good understanding of the full reconstruction chain and and event models.
  - The idea that user needs “just a script” is absolutely stupid.
  - Algorithm for the primary vertex reconstruction have never been discussed!
  - No convenient way of navigation through different detectors types.
  - Development when one person takes responsibility for everything.
  - Fast simulation → digis?
  - Performance.

Legacy code



barrel

SpdEcalB.cxx

SpdEcalB.h

SpdEcalBPoint.cxx

SpdEcalBPoint.h

~~SpdEcalTB.cxx~~

~~SpdEcalTB.h~~

~~SpdEcalTB2.cxx~~

~~SpdEcalTB2.h~~

~~SpdEcalTB2HitProducer.cxx~~

~~SpdEcalTB2HitProducer.h~~

~~SpdEcalTB2Point.cxx~~

~~SpdEcalTB2Point.h~~

SpdEcalTB2Reconstruction.cxx

SpdEcalTB2Reconstruction.h

SpdEcalTBHitProducer.cxx

SpdEcalTBHitProducer.h

SpdEcalTBParSet.cxx

SpdEcalTBParSet.h

~~SpdEcalTBPoint.cxx~~

~~SpdEcalTBPoint.h~~

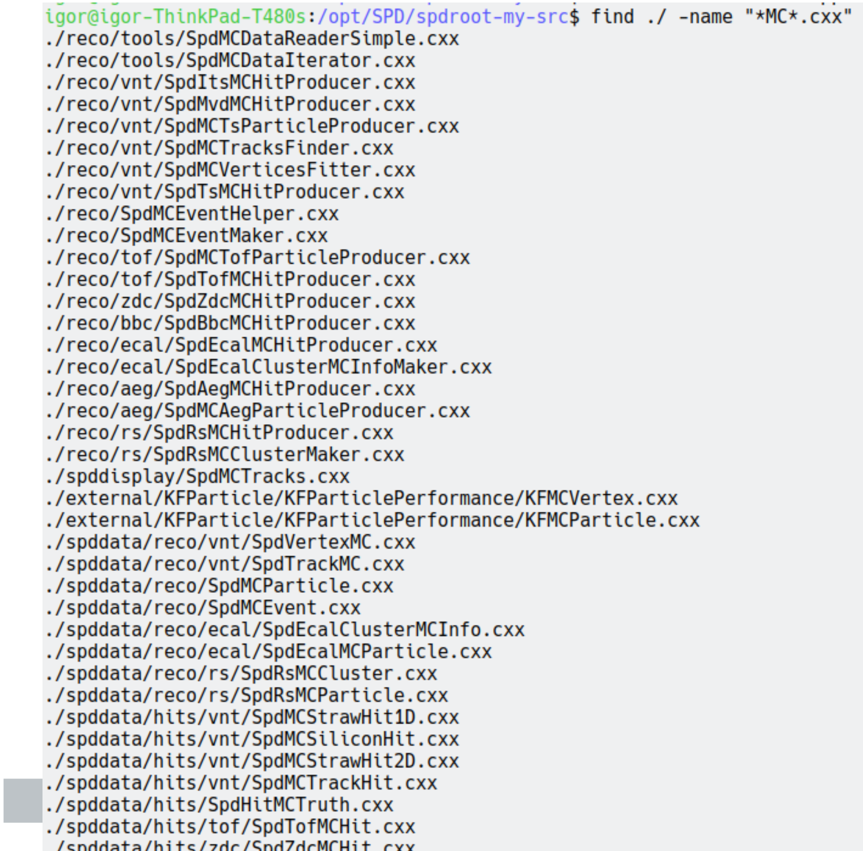
Some of other files may be removed, but it should be carefully checked!

There is orphaned code for RS and redundant geometrical options tracker not mentioning the simulation macros.

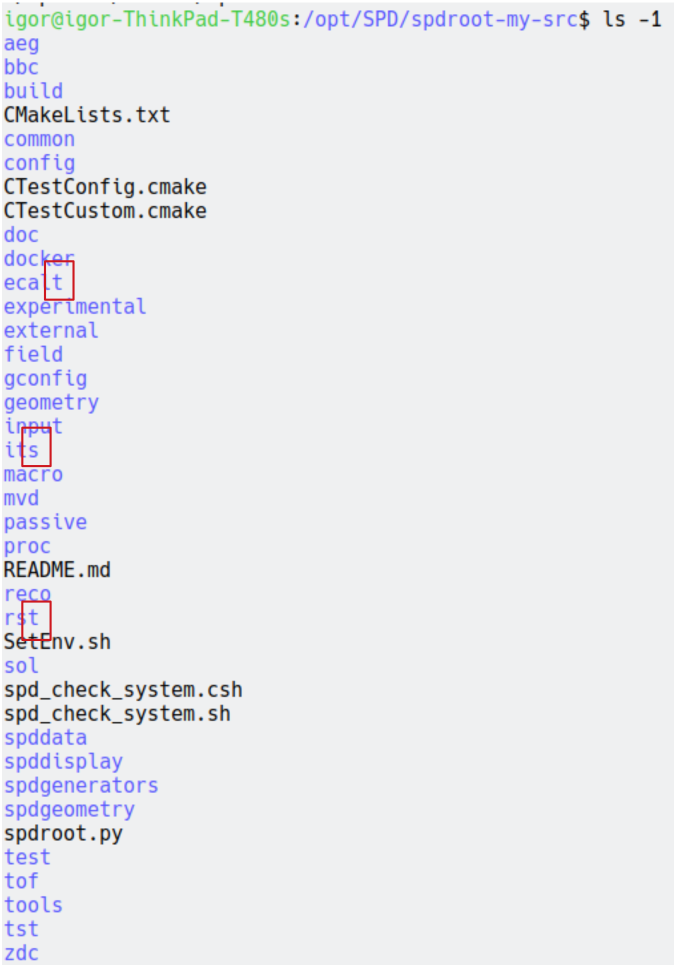
We can always take old code from git!

Naming conventions

- Subsystem names are not consistent with TDR!
- Legacy names referring to hybrid and quasi-solenoidal geometries (QSL, Q, …).
- “MC” is extensively used (e.g. SpdTrackMC and SpdMCTrack)



```
igor@igor-ThinkPad-T480s:/opt/SPD/spdroot-my-src$ find ./ -name "*MC*.cxx"
./reco/tools/SpdMCDataReaderSimple.cxx
./reco/tools/SpdMCDataIterator.cxx
./reco/vnt/SpdItsMCHitProducer.cxx
./reco/vnt/SpdMvdMCHitProducer.cxx
./reco/vnt/SpdMCTsParticleProducer.cxx
./reco/vnt/SpdMCTracksFinder.cxx
./reco/vnt/SpdMCVerticesFitter.cxx
./reco/vnt/SpdTsMCHitProducer.cxx
./reco/SpdMCEventHelper.cxx
./reco/SpdMCEventMaker.cxx
./reco/tof/SpdMCTofParticleProducer.cxx
./reco/tof/SpdTofMCHitProducer.cxx
./reco/zdc/SpdZdcMCHitProducer.cxx
./reco/bbc/SpdBbcMCHitProducer.cxx
./reco/ecal/SpdEcalMCHitProducer.cxx
./reco/ecal/SpdEcalClusterMCInfoMaker.cxx
./reco/aeg/SpdAegMCHitProducer.cxx
./reco/aeg/SpdMCAegParticleProducer.cxx
./reco/rs/SpdRsMCHitProducer.cxx
./reco/rs/SpdRsMCClusterMaker.cxx
./spddisplay/SpdMCTracks.cxx
./external/KFParticle/KFParticlePerformance/KFMCVertex.cxx
./external/KFParticle/KFParticlePerformance/KFMCParticle.cxx
./spddata/reco/vnt/SpdVertexMC.cxx
./spddata/reco/vnt/SpdTrackMC.cxx
./spddata/reco/SpdMCParticle.cxx
./spddata/reco/SpdMCEvent.cxx
./spddata/reco/ecal/SpdEcalClusterMCInfo.cxx
./spddata/reco/ecal/SpdEcalMCParticle.cxx
./spddata/reco/rs/SpdRsMCCluster.cxx
./spddata/reco/rs/SpdRsMCParticle.cxx
./spddata/hits/vnt/SpdMCStrawHit1D.cxx
./spddata/hits/vnt/SpdMCSiliconHit.cxx
./spddata/hits/vnt/SpdMCStrawHit2D.cxx
./spddata/hits/vnt/SpdMCTrackHit.cxx
./spddata/hits/SpdHitMCTruth.cxx
./spddata/hits/tof/SpdTofMCHit.cxx
./spddata/hits/zdc/SpdZdcMCHit.cxx
```



```
igor@igor-ThinkPad-T480s:/opt/SPD/spdroot-my-src$ ls -1
aeg
bbc
build
CMakeLists.txt
common
config
CTestConfig.cmake
CTestCustom.cmake
doc
docker
ecalt
experimental
external
field
gconfig
geometry
input
its
macro
mvd
passive
proc
README.md
reco
rst
SetEnv.sh
sol
spd_check_system.csh
spd_check_system.sh
spddata
spddisplay
spdgenerators
spdgeometry
spdroot.py
test
tof
tools
tst
zdc
```

Design problems

- We have SpdTrackMC and SpdTrackRC and algos mostly using the first one. Neither of them can be used in the full reconstruction chain. It means that all user-analysis scripts should be rewritten later.
- In winter we had discussions on having 4 tools to reconstruct vertices:
  - SpdMCVerticesFinder,
  - SpdRCVerticesFinder,
  - SpdRCKFpartV0Finder,
  - SpdMCVerticesFinder2.
- This is a road to nowhere…
- We should develop event model consistent with the full chain of reconstruction, which is opposite to the current structure intended to obtain quick basic results for CDR/TDR.

Geometry

- Geometry of a subsystem is constructed by “Builder” class based on the parameters stored in Builder, SpdCommonGeoMapper, detector specific geo-mapper class (distributed between three files!).
- The parameters are stored and used to rebuild the geometry in “reco”-script.
- Now we have many unused classes for **obsolete subsystems design/geometry** and **numerical “geotype” options**, which require long-time code reading and testing to understand what they mean.
- Terminology in the code is not consistent with TDR (e.g. ladder in DSSD).
- A major clean-up is needed.
- For things like planar detectors navigation tool would be nice (u and v directions)). What abot other detectors like MCT?
- If “digies” are implemented, how to navigate detector/tube/strip positions?

Summary

- SpdRoot allowed us to obtained all numbers required for TDR, but there are problems with its future development.
- SpdRoot will remain our main tool for detector simulation for a considerable time. We should use it to find the optimal event model and reconstruction chain.
- A major clean-up of the SpdRoot code is proposed along with moving to consistent naming scheme.
- As compared to the previous experience, we should decrease the time of adoption of new commits by providing automated tests and not putting responsibility to a single person. The code should be as modular and independent of FairSoft/FairRoot as possible. As our software is basically a prototype, our focus should be on understanding how things should work in general.
- Geometry description should be simplified (all parameters in one place, consistent names, separated from the main code), all options described in TDR should be implemented, the rest – removed. The geometry should be stored in standard format in case we change the framework.
- Navigation for geometry?
- The simulation → reconstruction chain to be replaced by generation → simulation → reconstruction.
- Can we expect a batch system, which would allow as to submit/use containers?