Overview of software activities

A. Sarti & C. Finck

IX FOOT Collaboration Meeting

DIPARTIMENTO DI SCIENZE
DI BASE E APPLICATE
PER L'INGEGNERIA

SAPIENZA
UNIVERSITÀ DI ROMA

Steps towards next data taking..

➡ Covid is not an excuse for us ! 🤪

➡ Last months have been devoted to the development of the software framework in view of the next data taking that will soon happen either in CNAO or GSI

   – Most of the changes were 'transparent' for the user and have been performed in order to allow that the users can run the software in an easy and flexible / reliable way

   – The same coding conventions have been adopted in all the core software libraries, allowing an easier 'reading' of the code for newbies. Geometry, channel mappings, detector configurations, data objects, setters, getters are constantly under review to improve the consistency among the different packages… **A LOT OF WORK**!

➡ We're going to focus on the 'non transparent' changes that happened!

Being flexible..

➡ How to provide to the user a tool for:

– Producing the .inp for fluka / input for G4 simulations

– 'tupling' the .txt output for fluka

– Decoding the tupled output / Decoding the raw data (in a transparent way!) [level0 processing]

– Running the global event reconstruction on the output of level0 decoding

➡ Implemented a base class (TAGfoot/Base*) with all the key ingredients already prepared, with proper setting of input mapping, calibrations, geometry infos, handling different possibile input infos

➡ Implemented few classes that perform the 'selection' of the needed blocks accordingly to the task (MC geo creation, tupling, decoding, reconstruction): Local* and Global* classes in TAGfoot

Of course, everything relies on the base class for all the detectors, and subsystems.. hidden from the user

..handling different data!

➡ But how to customise the job accordingly to the users needs?

– Want to track? Want to decode the hits? Decode raw data? MC? Run the global reconstruction? Perform calibration? Study only VTX & TW?

– Want to study GSI MC? Data? CNAO2020? Oxygen MC? etc. etc..

– Which runs? Exp. conditions can change even within a given data taking campaign…

➡ The implemented solution:

– **Campaign Manager + Run info**. The users needs to:

  • Specify via command line the 'campaign', run and data type that wants to process: -exp GSI -run 2291 (data) or -exp GSI -run 1 -mc (MC)

  • Verify that the setting in the corresponding configuration file are matching his needs: config/GSI/FootGlobal.par

  • Verify that the campaign/run pair that is under study is 'available' in the campaign manager…cammaps/GSI.cam

A lot happens 'behind'

```
Known Actions:
  name                type
  locRecFile          TAGactTreeWriter
  actGeoTrafo         TAGgeoTrafo
  actCamMan           TAGcampaignManager
  actEvtReader        TAGactTreeReader
  eveActNtuMc         TAMCactNtuEve
  stActNtu            TASTactNtuMC
  stActNtuMc          TAMCactNtuStc
  bmActNtu            TABMactNtuMC
  bmActNtuMc          TAMCactNtuBm
  vtActNtu            TAVTactNtuMC
  vtActNtuMc          TAMCactNtuVtx
  itActNtu            TAITactNtuMC
  itActNtuMc          TAMCactNtuItr
  msdActNtu           TAMSDactNtuMC
  msdActNtuMc         TAMCactNtuMsd
  twActNtu            TATWactNtuMC
  twActNtuMc          TAMCactNtuTof
  caActNtu            TACAactNtuMC
  caActNtuMc          TAMCactNtuCal
  bmActTrack          TABMactNtuTrack
  vtActClus           TAVTactNtuClusterF
  vtActTrack          TAVTactNtuTrackF
  vtActVtx            TAVTactNtuVertexPD
  itActClus           TAITactNtuClusterF
  msdActClus          TAMSDactNtuCluster
  twActPoint          TATWactNtuPoint
  caActClus           TACAactNtuCluster
```

```
Known ParaDsc's:
  name                type
  tgGeo               TAGparGeo
  trGeo               TASTparGeo
  bmGeo               TABMparGeo
  bmConf              TABMparConf
  diGeo               TADIparGeo
  vtGeo               TAVTparGeo
  vtConf              TAVTparConf
  itGeo               TAITparGeo
  itConf              TAITparConf
  msdGeo              TAMSDparGeo
  msdConf             TAMSDparConf
  twGeo               TATWparGeo
  twCal               TATWparCal
  caGeo               TACAparGeo
```

```
Known DataDsc's:
  name       type              produced by
  eveMc      TAMCntuEve        eveActNtuMc
  stRaw      TASTntuRaw        stActNtu
  stMc       TAMCntuHit        stActNtuMc
  bmRaw      TABMntuRaw        bmActNtu
  bmMc       TAMCntuHit        bmActNtuMc
  vtRaw      TAVTntuRaw        vtActNtu
  vtMc       TAMCntuHit        vtActNtuMc
  itRaw      TAITntuRaw        itActNtu
  itMc       TAMCntuHit        itActNtuMc
  msdRaw     TAMSDntuRaw       msdActNtu
  msdMc      TAMCntuHit        msdActNtuMc
  twRaw      TATWntuRaw        twActNtu
  twMc       TAMCntuHit        twActNtuMc
  caRaw      TACAntuRaw        caActNtu
  caMc       TAMCntuHit        caActNtuMc
  bmTrack    TABMntuTrack      bmActTrack
  vtTrack    TAVTntuTrack      vtActTrack
  vtVtx      TAVTntuVertex     vtActVtx
  vtClus     TAVTntuCluster    vtActClus
  itClus     TAITntuCluster    itActClus
  msdClus    TAMSDntuCluster   msdActClus
  twPoint    TATWntuPoint      twActPoint
  caClus     TACAntuCluster    caActClus
```

```
Required Actions:
  actEvtReader        TAGactTreeReader
  eveActNtuMc         TAMCactNtuEve
  stActNtuMc          TAMCactNtuStc
  bmActNtuMc          TAMCactNtuBm
  vtActNtuMc          TAMCactNtuVtx
  itActNtuMc          TAMCactNtuItr
  msdActNtuMc         TAMCactNtuMsd
  twActNtuMc          TAMCactNtuTof
  caActNtuMc          TAMCactNtuCal
  locRecFile          TAGactTreeWriter
  stActNtu            TASTactNtuMC
  bmActNtu            TABMactNtuMC
  bmActTrack          TABMactNtuTrack
  vtActNtu            TAVTactNtuMC
  vtActClus           TAVTactNtuClusterF
  vtActTrack          TAVTactNtuTrackF
  vtActVtx            TAVTactNtuVertexPD
  itActNtu            TAITactNtuMC
  itActClus           TAITactNtuClusterF
  msdActNtu           TAMSDactNtuMC
  msdActClus          TAMSDactNtuCluster
  twActNtu            TATWactNtuMC
  twActPoint          TATWactNtuPoint
  caActNtu            TACAactNtuMC
  caActClus           TACAactNtuCluster
```

The job control…

```
IncludeKalman:    n
IncludeTOE:       n
EnableLocalReco: n          IncludeDI:               y
                            IncludeST:               y
EnableTree:       y         IncludeBM:               y
EnableHisto:      y         IncludeTG:               y
EnableTracking:   y         IncludeVT:               y
                            IncludeIT:               y
EnableSaveHits:   n         IncludeMSD:              y
EnableRootObject: n         IncludeTW:               y
EnableTofZmc:     n         IncludeCA:               y
EnableTofCalBar:  n
```

.. examples (II)

➡ Proper tuning of all the actions is ensured by the campaign manager.

➡ All the relevant files are paired to the campaign and run of interest inside the cammaps folder.

    ➡ Best example to have a look at: cammaps/ GSI.cam

➡ Info about what has been decoded and is available for further analysis/inspection is kept inside the TAGrunInfo object

```
KEY: TAGrunInfo runinfo;1
root [3] runinfo->Print()
Run info:        cam: 12C_200/  run:     1
Global info:
  EnableLocalReco: 0
  EnableTree: 1
  EnableHisto: 1
  EnableTracking: 1
  EnableSaveHits: 0
  EnableRootObject: 0
  EnableTofZmc: 0
  EnableTofCalBar: 0

  IncludeKalman: 0
  IncludeTOE: 1

  IncludeDI: 1
  IncludeST: 1
  IncludeBM: 1
  IncludeTG: 1
  IncludeVT: 1
  IncludeIT: 1
  IncludeMSD: 1
  IncludeTW: 1
  IncludeCA: 0
```

➡ Details can be found in the software twiki pages.

    – The campaign manager details are described in few talks from C. Finck

    – Examples are outlined inside the software page

Typical 'sequence'

➡ Quickstart guide:

- Follow the instructions on the wiki page and install/compile the software kept in the Master branch

- Identify your favourite run/data type/campaign

- go inside the build/Simulation or build/Reconstruction/ {level0,fullrec} folders **and run your favourite executable**..

➡ Should take really few minutes and you'll be able to…

```
../../bin/DecodeMC -in data/12C_C_200_1.root -exp 12C_200 -run 1 -out testMCDec.root -mc
```

➡ .. that usually results quickly in a fantastic…… ntuple? histogram?

```
*** Break *** segmentation violation
[/usr/lib/system/libsystem_platform.dylib] _sigtramp (no debug info)
[<unknown binary>] (no debug info)
```
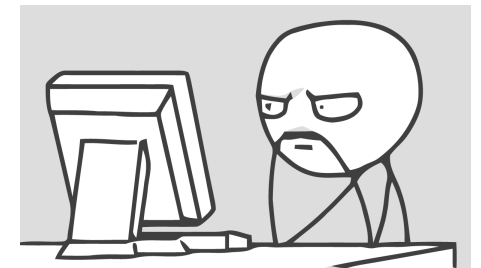
.. and then the fun begins!

➡ Before going down the path of extreme actions..

➡ Debug is far more easy if:

  – You have a close look at the error message after the crash :) [usually provides hints..]

  – You check the job input! Are you trying to process MC without specifying the -mc flag? Trying to access a non existing campaign/run pair (eg: -exp CNAO2019 or -exp GSI -run 0.23)?

  – You check that inside FootGlobal.par under the campaign you are looking for, the options are properly setup:

    • If I am running against the GSI campaign, I will fail to access the Calo info that is not there!

➡ Ok,ok… it's true that we should try to exit with something better than a 'segmentation violation' and give to the user some more clearer info about what's going on… **But for that there's the foot-software-develop@lists.infn.it mailing list :)**
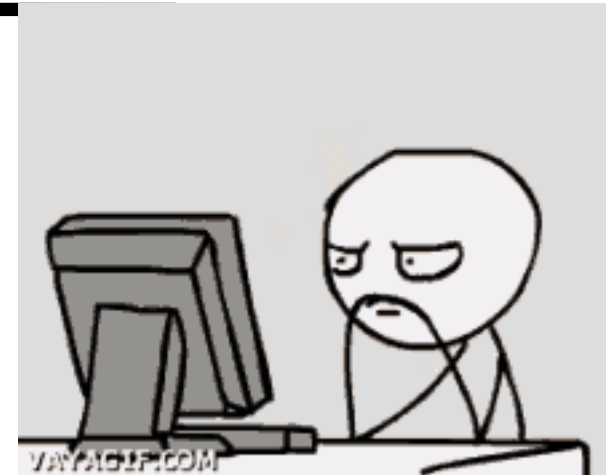
Subsystems report I

➡ Simulation

- FLUKA [More info in the next talk from Giuseppe.]
  - Calo geometry visualisation and modularity implementation in the general framework is ongoing (needs manpower for flair!)
- Campaign managers handling still to be checked in detail
- Geant4 / FLUKA: we need to discuss - plan future studies and eventually publications. Discuss strategies to measure X-Sections, Efficiencies, high statistics sample production and storage…

➡ SC ready for data taking

➡ BM ready for data taking. S-T calibration / implementation being improved

➡ VTX + IT ready for data taking

➡ MSD: to be developed (templates are there, but we need the actual implementation)

Subsystems report II

➡ TW: DeltaE, ToF calibration is being implemented in shoe. Ready for data taking.

➡ Calo : DAQ to be developed using the SC and TW templates. Will happen soon after Christmas break. Clustering algorithm being developed (see talk from F. Cavanna in this meeting).

➡ DAQ : everything should be ok. Need 'integrated' setup / data to test!

➡ Analysis

   – **We finally have a running global reconstruction that produces an ntuple with the reconstructed momentum of all the tracks!**

   – Will be the basis for developing the tools for data analysis and X-section calculations using as input reconstructed quantities!
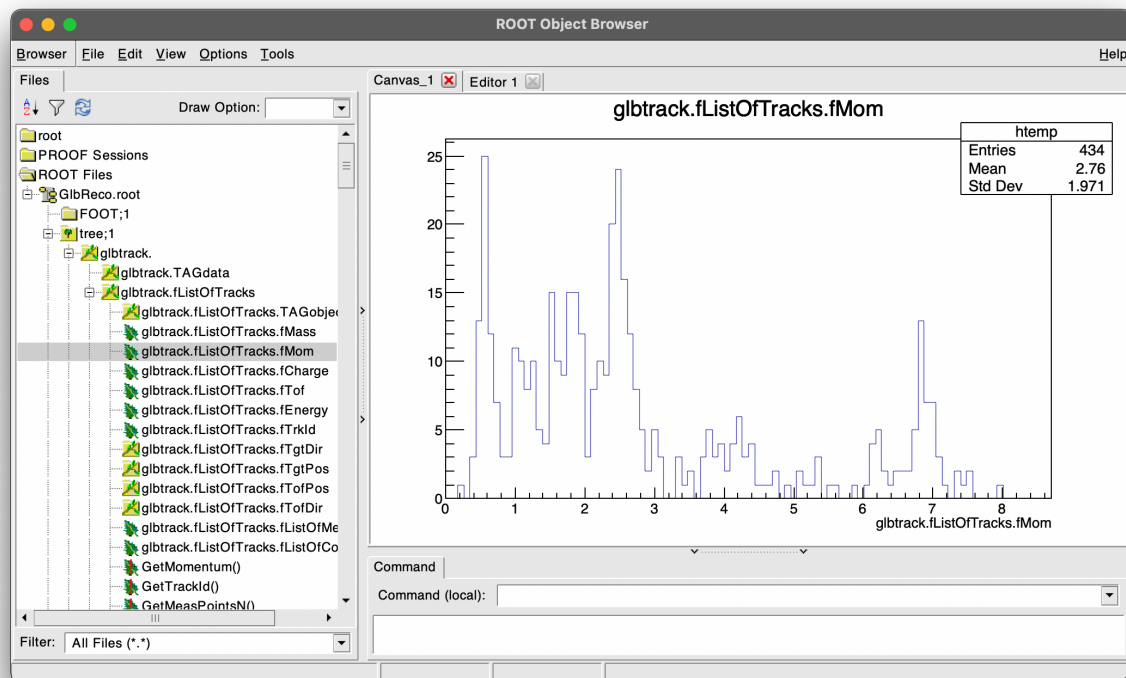
Reconstruction

➡ Level 0

- Finalise the 'MC processing strategy' to improve speed, avoid storing 'trees' that have to be converted in TAG* objects, overcoming the 'tree' limitations that come from using fixed length arrays…

- **Improve the code speed / documentation**.. mandatory to have a fast feedback during the real data taking!

  - Still few options open for investigation: divide the L0 decoding in two parts (e.g. WD decoding done once to spare time,…),

- Evaluate if there's an easy way to implement few relevant plots inside **GNAM to have 'online' feedback** instead of waiting for an event to be stored/built/shipped for the offline reconstruction

➡ Global Reconstruction

- We just need to start to use it extensively for our studies! There's a lot of work to do in refining the strategies related to 'putting everything together'

A powerful tool

➡ The 'flexible' structure allows the users to customise the 'main' (executable) according to their favourite needs!

    – E.g. Roberto was able to embed the TW standalone calibration tools inside the shoe framework without suffering too much!

➡ The main 'take home message' is: there are no excuses, right now, for not using it :)
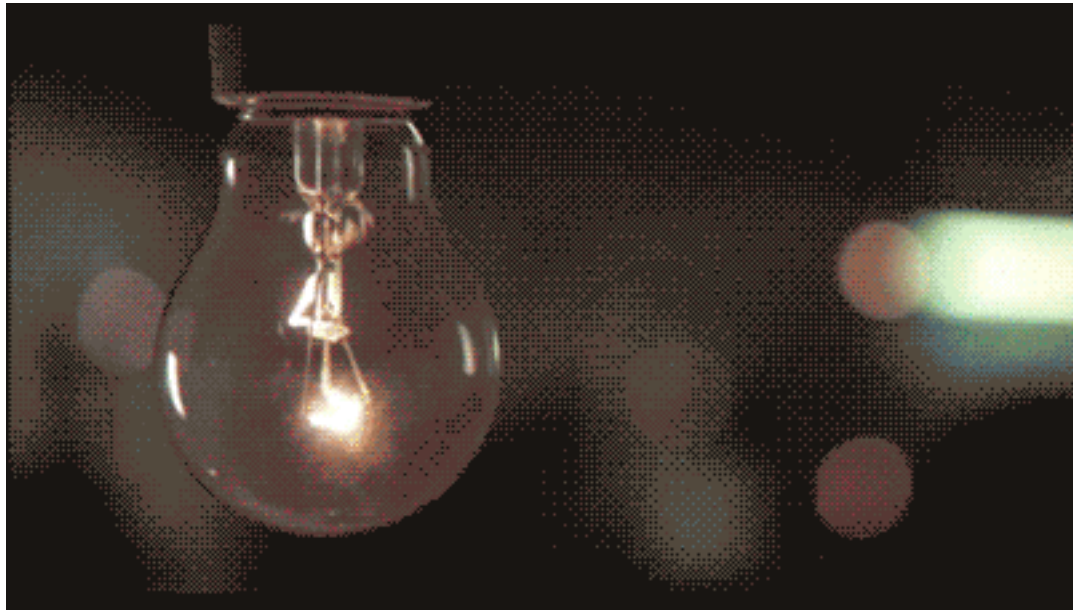
…still plenty of work to be done

➡ As the general framework is getting more and more stable/harmonised, we need the sub-detector experts to really start to contribute to the software project.

- Geometry, mapping, calibration information has to be kept up-to-date and validated by the experts for EACH campaign!
  - This, actually is a job much 'tougher' than it seems! **Most of the time we get the infamous 'Segmentation violation' because some of the input info for the setup of a sub-detector reconstruction is missing or wrong**… Protections should be provided inside the code: if a file is missing or the info cannot be loaded an 'understandable' error message should be provided by the software!
- Experts have to follow up closely the Master branch releases implementation, receive in their development branches the latest developments and ensure that their classes/algorithms are properly updated..
  - John 1:23 "I am the voice of one crying in the wilderness, Make straight the way of the Lord"

The near future..

➡ To spot problems and to ease the interface for the users we need…

    – **Users! (And feedback from them..)**

➡ Almost everything I have shown and discussed there is useless unless… .. Unless someone (besides me, Marco, Lorenzo/Francesca and Chris/Alexandre) really starts to use the software to look at the data/MC simulation performing a global reconstruction of the events…

➡ To ease the access of people to the software we plan to organise a **shoe software hands on tutorial in Jan 2021**.

    – A doodle will be circulated in the foot-general mailing list to identify the best couple of days for this 'online' course in which people will be introduced to the magic world of shoe… and his easy to spell objects and classes like (TAVTactBaseNtuClusterMT)

    – Subsystem experts presence will be **mandatory and enforced** [violence has been allowed by the spokesman] :)