Campaign Manager

Master File

Files

Implementation

Reconstruction

Local Reco executable

Conclusions

Master File

- Structure: ./cammaps/FOOT_cam.map

```
// List of Campaigns
CamNumber:    0
CamName:      ""
CamDataMC:    1
CamDate:      ""
CamSum: "Test using standard input files"

CamNumber:    1
CamName:    "GSI"
CamDataMC:    0
CamDate: "5-8 April 2019"
CamSum: "Test experiment with ST+BM+VTX+TW…"

CamNumber:    2
CamName:   "HE_MC"
CamDataMC:    1
CamDate: "5-8 January 2020"
CamSum: "MC data for HE reaction produced by FLUKA"
```

- Campaign name with associated id
- MC data flag
- Date of data taking or production
- Summary of the campaign

➡ e.g.: load campaign file in ./cammaps/GSI.cam when option: *-exp* GSI

File (i)

- Real data structure: ./cammaps/GSI.cam

```
// Campaign file
CamName: "GSI"
RunNumber: 2210-2212; 2238-2242; 2251-2254
NumberDevices: 6

DetectorName: "FOOT"
NumberFiles: 1
“./geomaps/GSI/FOOT.geo”: 2210; 2251

DetectorName: "ST"
NumberFiles: 4
"./geomaps/GSI/TASTdetector.geo": -1
“./config/GSI/TASTdetector.cfg”: -1
"./config/GSI/WDChannelMap.txt": -1
"./config/GSI/WDTimeCalibration/tcalib": 2210; 2251

DetectorName: "BM"
NumberFiles: 3
"./geomaps/GSI/TABMdetector.geo": -1
"./config/GSI/TABMdetector.cfg": -1
"./config/GSI/T0_beammonitor.cfg": 2210; 2251

DetectorName: "TG"
NumberFiles: 1
"./geomaps/GSI/TAGdetector.geo": -1

DetectorName: "VT"
NumberFiles: 2
“./geomaps/GSI/TAVTdetector.geo": -1
"./config/GSI/TAVTdetector.cfg": -1
```

```
DetectorName: "TW"
NumberFiles: 2
“./geomaps/GSI/TATWdetector.geo": -1
"./config/GSI/TATWChannelMap.xml": -1
```

- Campaign name with associated run
- Detector name with associated file number and name
- Name following with the corresponding run number validity (-1 no dependence)

➡ File should not be modified (unless for error correction)

➡ e.g.: config/GSI/T0_beammonitor_2210.cfg (works also for folders)

File (ia)

- Run dependency: (as example)

```
// Campaign file
CamName: "GSI"
RunNumber: 2210-2212; 2238-2242; 2251-2254
NumberDevices: 6

. . .

DetectorName: "BM"
NumberFiles: 3
"./geomaps/GSI/TABMdetector.map": -1
"./config/GSI/TABMdetector.cfg": -1
"./config/GSI/T0_beammonitor.cfg": 2210; 2251
. . .
```

- Config file ./config/GSI/T0_beammonitor_2210.cfg for BM is valid for runs 2210-2212 and 2238-2242

- Config file ./config/GSI/T0_beammonitor_2251.cfg for BM is valid for runs 2251-2254

File (ii)

- MC Structure: ./cammap/HE_MC.cam

```
/ Campaign file
CamName: "HE_MC"
RunNumber: 1-3
NumberDevices: 9

DetectorName: "FOOT"
NumberFiles: 1
“/geomaps/HE_MC/FOOT.geo”: -1

DetectorName: "ST"
NumberFiles: 5
"./geomaps/HE_MC/TASTdetector.geo": -1
"./config/HE_MC/TASTdetector.cfg": -1
"./config/HE_MC/TATWChannelMap.xml": -1
"./config/HE_MC/WDChannelMap.txt": -1

DetectorName: "BM"
NumberFiles: 3
"./gemaps/HE_MC/TABMdetector.geo": -1
"./config/HE_MC/TABMdetector.cfg": -1
"./config/HE_MC/T0_beammonitor.cfg": -1

DetectorName: "TG"
NumberFiles: 1
"./geomaps/HE_MC/TAGdetector.geo": -1

DetectorName: "VT"
NumberFiles: 2
“./geomaps/HE_MC/TAVTdetector.geo": -1
"./config/HE_MC/TAVTdetector.cfg": -1
```

```
DetectorName: "IT"
NumberFiles: 2
"./geomaps/HE_MC/TAITdetector.geo": -1
"./config/HE_MC/TAITdetector.cfg": -1

DetectorName: "MSD"
NumberFiles: 1
"./geomaps/HE_MC/TAMSDdetector.geo": -1

DetectorName: "TW"
NumberFiles: 1
"./geomaps/HE_MC/TATWdetector.geo": -1

DetectorName: "CA"
NumberFiles: 1
“./geomaps/HE_MC/TACAdetector.geo": -1
```

➡ For standard (e.g. MC) all detectors present.

➡ Campaign manager will return the name of the parameter files with the right name with respect to campaign and run number for a given detector.

Implementation (i)

- TAGcampaignManager:

```
class TAGcampaignManager : public TAGaction {
struct CamParameter_t : public  TNamed {
      TString   Name;      // Campaign name
      Int_t     Number;    // Campaign number
      Bool_t    McFlag;    // Flag for MC data (0 for real data)
      TString   Date;      // Date of data taking or production
      TString   Summary;   // Summary of the campaign
   };
   TAGparTools*   fFileStream;
   TAGcampaign*   fCurCampaign;
   Int_t          fCurCampaignNumber;
   CamParameter_t fCamParameter[128];

public:
   TAGcampaignManager(const TString exp = "");
   virtual ~TAGcampaignManager();

   Bool_t FromFile(TString ifile = "");

   Int_t               GetCurrentCamNumber() const { return fCurCampaignNumber; }
   CamParameter_t&     GetCampaignPar(Int_t idx)   { return fCamParameter[idx]; }
   const TAGcampaign*  GetCurCampaign()            { return fCurCampaign;       }

   const Char_t*       GetCurGeoFile( const TString&  detName, Int_t runNumber = -1);
   const Char_t*       GetCurConfFile(const TString&  detName, Int_t runNumber = -1);
   const Char_t*       GetCurMapFile( const TString&  detName, Int_t runNumber = -1);
   const Char_t*       GetCurCalFile( const TString&  detName, Int_t runNumber = -1);
   Bool_t              IsDetectorOn(  const TString&  detName)
. . .
};
```

➡ Handles all campaigns and gives back information about the current campaign set by “exp” argument

Implementation (ii)

- TAGcampaign:

```
class TAGcampaign : public TAGparTools {
public:
   TAGcampaign();
   virtual ~TAGcampaign();

   Bool_t            FromFile(TString ifile = "");
   const Char_t*     GetName()     const { return fName.Data(); }
   const TArrayI&    GetRunArray() const { return fRunArray;    }
   Int_t             GetDevicesN() const { return fDevicesN;    }

   const Char_t*     GetGeoFile(const  TString& detName, Int_t runNumber);
   const Char_t*     GetConfFile(const TString& detName, Int_t runNumber);
   const Char_t*     GetMapFile(const  TString& detName, Int_t runNumber);
   const Char_t*     GetCalFile(const  TString& detName, Int_t runNumber);

   Bool_t            IsDetectorOn(const TString& detName);

. .
   ClassDef(TAGcampaign,1)
};
```

➡ Campaign class, stores names of the parameter file for a given detector name and run number

Implementation (iii)

- TAGcampaign:

```
//_____________________________________________________________________________
Bool_t TAGcampaign::CheckFiles()
{
   for ( map<TString, TString>::const_iterator it = fFileGeoMap.begin(); it != fFileGeoMap.end(); ++it) {
      const Char_t* name = it->second;

      if( access(name, F_OK) == -1 ) {
         Warning("CheckFiles()", "File %s not found !", name);
         return false;
      }
   }
. . .
   for ( map<TString, TString>::const_iterator it = fFileCalMap.begin(); it != fFileCalMap.end(); ++it) {
      const Char_t* name = it->second;

      if( access(name, F_OK) == -1 ) {
         Warning("CheckFiles()", "File %s not found !", name);
         return false;
      }
   }

   return true;
}
```

➡Campaign class checks presence of all files (geo, cal, cfg, map)

Implementation (iv)

- Printouts: masterfile (i)

```
Number of campaigns: 3
  Campaign number:  0
  Campaign name:
  MC data flag:     1
  Campaign period:
  Campaign summary: Test using standard input files

  Campaign number:  1
  Campaign name:    GSI
  MC data flag:     0
  Campaign period:  5-8 April 2019
  Campaign summary: Test experiment with ST+BM+VTX+TW…

  Campaign number:  2
  Campaign name:    HE_MC
  MC data flag:     1
  Campaign period:  5-8 January 2020
  Campaign summary: MC data for HE reaction produced by FLUKA
```

Implementation (v)

- Printouts: GSI (ii)

```
Geometry files for GSI:
  Device name: BM with file: ./geomaps/GSI/TABMdetector.geo
  Device name: FOOT with file: ./geomaps/GSI/FOOT.geo
  Device name: ST with file: ./geomaps/GSI/TASTdetector.geo
  Device name: TG with file: ./geomaps/GSI/TAGdetector.geo
  Device name: TW with file: ./geomaps/GSI/TATWdetector.geo
  Device name: VT with file: ./geomaps/GSI/TAVTdetector.geo
  Device name: BM  Run number:-1
  Device name: FOOT  Run number:2210 2251
  Device name: ST  Run number:-1
  Device name: TG  Run number:-1
  Device name: TW  Run number:-1
  Device name: VT  Run number:-1
Configuration files for GSI:
  Device name: BM with file: ./config/GSI/TABMdetector.cfg
  Device name: ST with file: ./config/GSI/TASTdetector.cfg
  Device name: VT with file: ./config/GSI/TAVTdetector.cfg
  Device name: BM  Run number:-1
  Device name: ST  Run number:-1
  Device name: VT  Run number:-1
Mapping files for GSI:
  Device name: BM with file: ./config/GSI/TABMdetector.map
  Device name: TW with file: ./config/GSI/TATWChannelMap.xml
  Device name: BM  Run number:-1
  Device name: TW  Run number:-1
Calibration files for GSI:
  Device name: BM with file: ./config/GSI/T0_beammonitor.cfg
  Device name: ST with file: ./config/GSI/WDTimeCalibration/tcalib/
  Device name: BM  Run number:2210 2251
  Device name: ST  Run number:2210 2251
```

➡ printout option “all”: gives all files with the valid run numbers (-1 no dependency)

Reconstruction (i)

- BaseReco (i):

```
class BaseReco : public TNamed // using TNamed for the in/out files
{
  . . .
protected:
  TAGcampaignManager* fCampManager;
  Int_t               fRunNumber;
. . .
};
```

```
//_____________________________________________________________
void BaseReco::ReadParFiles()
{
. . .
if (GlobalPar::GetPar()->IncludeTG()) {
     fpParGeoG = new TAGparaDsc(TAGparGeo::GetDefParaName(), new TAGparGeo());
     TAGparGeo* parGeo = (TAGparGeo*)fpParGeoG->Object();

     TString parFileName = fCampManager->GetCurGeoFile(TAGparGeo::GetBaseName(), fRunNumber);
     parGeo->FromFile(parFileName.Data());
}
. . .
```

➡ Campaign manager from detector name and the run number will compose the name of the different cfg, map, cal, geo files.

➡ Campaign manager implemented but not used for parameter file loading.

Reconstruction (ii)

- BaseReco (ii):

```
//_____________________________________________________________________________
void BaseReco::CampaignChecks()
{
   // check detector include in FootGlobal.par vs current campaign
   vector<TString> list = GlobalPar::GetPar()->DectIncluded();
   for (vector<TString>::const_iterator it = list.begin(); it != list.end(); ++it) {
      TString str = *it;

      if (!fCampManager->IsDetectorOn(str)) {
         Error("CampaignChecks()", "the detector %s is NOT referenced in campaign file", str.Data());
         exit(0);
      }
   }
   TArrayI runArray = fCampManager->GetCurRunArray();
   Bool_t runOk = false;

   for (Int_t i = 0; i < runArray.GetSize(); ++i) {
      if (fRunNumber == runArray[i])
         runOk = true;
   }
   if (!runOk) {
      Error("CampaignChecks()", "run %d is NOT referenced in campaign file", fRunNumber);
      exit(0);

   }
}
```

➡ Check the campaign name and run number vs database

➡ Check if detector in Global par is in list of the campaign

Local Reco executable

- DecodeRaw(MC)

```
int main (int argc, char *argv[])  {
. . .
   Int_t runNb = -1;

  for (int i = 0; i < argc; i++){
. . .
     if(strcmp(argv[i],"-run") == 0)   { runNb = atoi(argv[++i]);  }   // Run Number
  }
. . .
   LocalReco* locRec = new LocalReco(exp, in, out);
. . .
 if (runNb != -1)
      locRec->BaseReco::SetRunNumber(runNb);
. . .
}
```

➡ Add option for run number, if present, set it in local reco

Event display

- TAEDbaseInterface:

```
class TAEDbaseInterface : public TEveEventManager
{
. . .
   virtual void ShowDisplay(const TString fileName, Int_t runNumber = -1);
. . .
protected:
   TString            fExpName;
   Int_t              fType;
   Int_t              fRunNumber;
. . .
};
```

➡Add run number as member and in argument in ShowDisplay method

Conclusions (i)

- This structure will allow to make the association btw configuration/geometry/ calibration/mapping files with campaign name and run number
- Correct some bugs

➡ Need to have a coherent extension for all the files:

*.geo for geometry files in geomaps folder

*.map for mapping file in config folder

*. cfg for configuration files in config folder

*. cal for calibration files in calib folder

➡ not homogenous: e.g. .txt, .xml extension or calib file with .cfg extension, etc…. (folders are ok, should finish with a '/')

Conclusions (ii)

➡ Need different run numbers for a given campaign !

➡ Need some cleaning of old files !

- Will ease the readability of the code if we stick on some convention
- Campaign Manager manages all present detectors, GlobalPar allows to switch off/on given detectors during reconstruction.