

Data storage manager

New class

□ Element class (TAGelement)

- Manage MC hit/particle array and relation array

```
class TAGelement : public TAGobject {
public:
    TAGelement();
    virtual ~TAGelement();

    //! return MC hit index
    Int_t      GetMcIndex(Int_t index)      const { return fMcIndex[index]; }
    //! return MC track index
    Int_t      GetMcTrackIdx(Int_t index)   const { return fMcTrackIdx[index]; }
    //! return number of MC hits
    Int_t      GetMcIndexesN()             const { return fMcIndex.size(); }
    //! return number of MC tracks
    Int_t      GetMcTracksN()              const { return fMcTrackIdx.size(); }

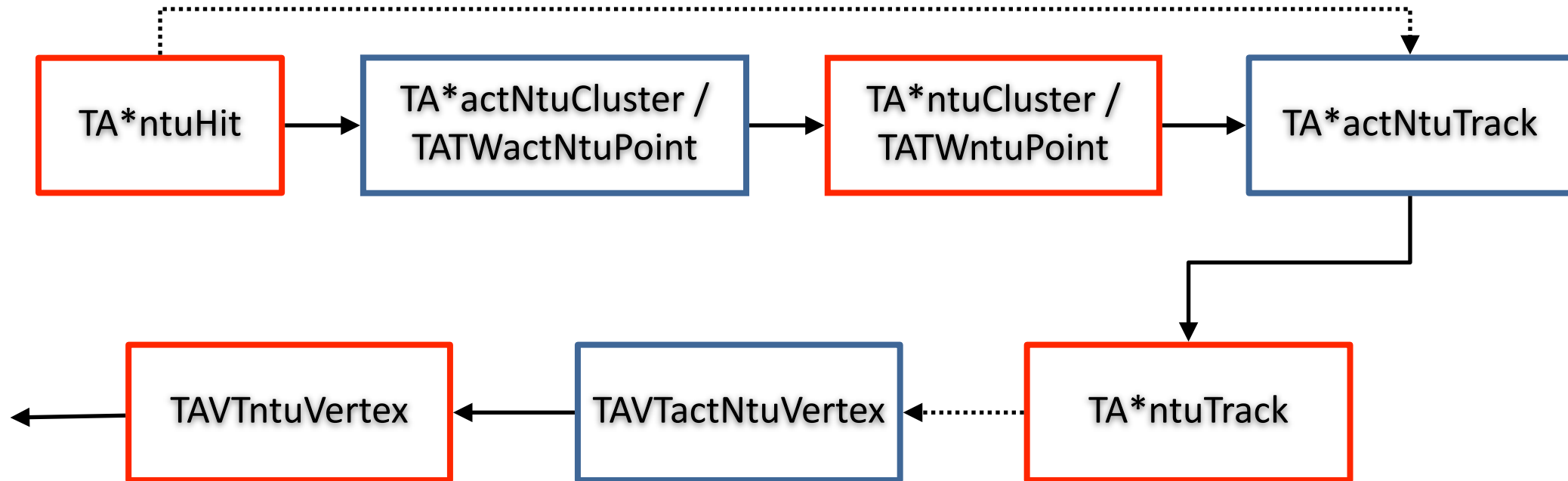
    //! Get relation first index
    Int_t      GetRelationFirst(Int_t idx)  const { return TAGparTools::GetPairFirst(fRelation[idx]); }
    //! Get relation second index
    Int_t      GetRelationSecond(Int_t idx) const { return TAGparTools::GetPairSecond(fRelation[idx]); }
    //! Get relation index number
    Int_t      GetRelationsN()              const { return fRelation.size(); }

    // add MC info
    void       AddMcTrackIdx(Int_t trackIdx, Int_t mcId = -1);
    // Add a relation
    void       AddRelation(Int_t sensorId, Int_t index = 0);
};
```

- ➔ All containers interfaced with this class
- ➔ Same implementation and checks duplicate
- ➔ Save line of code

Reconstruction

□ Scheme

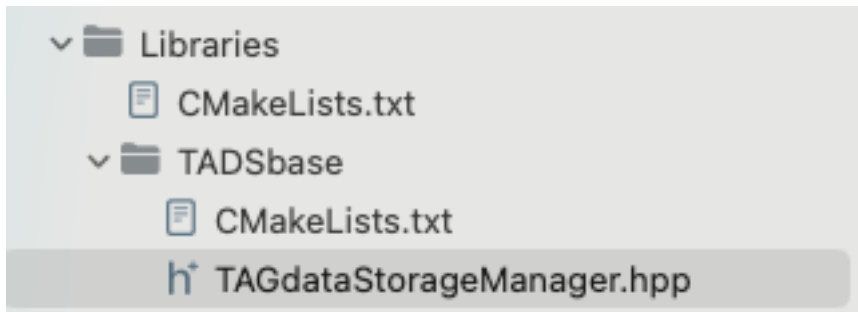


- For all clusters, the attached hits are saved
 - For all tracks, the attached clusters are saved
 - For all vertices, the attached tracks are saved
- ➔ Despite all objects already in the dedicated containers
- ➔ Instead of saving the objects, only save the indexes

New data storage manager (i)

□ TAGdataStorageManager:

- Template class, need to build an interface library in a new folder



➔ Update CMake files accordingly

New data storage manager (ii)

□ TAGdataStoredManager:

```
class TAGdataStoredManager : public TObject
{
public:
    . . .
    //-----+-----
    //! Get Clusters
    template <class NtuCluster, class Cluster, class Track>
    static vector<Cluster*> GetClusters(Track* track)
    {
        NtuCluster ntu;
        TString name = gTAGroot->DefaultDataDscName(ntu.Class()->GetName());
        TAGdataDsc* ntuClusDsc = gTAGroot->FindDataDsc(name.Data());

        vector<Cluster*> vec;

        if (ntuClusDsc) {
            NtuCluster* ntuClus = (NtuCluster*) ntuClusDsc->Object();

            for (Int_t i = 0; i < track->GetRelationsN(); ++i) {
                Int_t sensIdx = track->GetRelationFirst(i);
                Int_t clusIdx = track->GetRelationSecond(i);
                vec.push_back(ntuClus->GetCluster(sensIdx, clusIdx));
            }
            return vec;
        } else {
            printf("GetClusters: No relation found for %s", ntu.Class()->GetName());
            return vec;
        }
    }
    . . .
}
```

- ➔ Excerpt for retrieving cluster information from a given track
- ➔ Same holds true for hits from a given clusters or tracks from a given vertex

New data storage manager (iii)

□ TAGdataStorageManager:

- Retrieving object from original container
- Example: retrieving clusters from a track

```
TAGbaseTrack* track = NewTrack();
vector<TAVTcluster*> vec = TAGdataStoredManager::GetClusters<TAVTntuCluster, TAVTcluster>(track);
printf("size %d\n", (int)vec.size());

for (auto itr : vec) {
    TVector3 pos = itr->GetPosition();
    pos.Print();
}
```

➔ The method is static, can be called anywhere as long as the containers are defined !

➔ Returns a std vector with the clusters associated to the given track in argument

```
size 3
TVector3 A 3D physics vector (x,y,z)=( 0.264143,-0.114013,0.000000) (rho,theta,phi)=(0.287699,90.000000,-23.346737)
TVector3 A 3D physics vector (x,y,z)=(-0.297433,-0.112427,0.000000) (rho,theta,phi)=(0.317972,90.000000,-159.293888)
TVector3 A 3D physics vector (x,y,z)=(-0.263925,-0.127305,0.000000) (rho,theta,phi)=(0.293024,90.000000,-154.249525)

size 3
TVector3 A 3D physics vector (x,y,z)=( 0.205505,-0.113850,0.000000) (rho,theta,phi)=(0.234934,90.000000,-28.986502)
TVector3 A 3D physics vector (x,y,z)=(-0.241451,-0.112963,0.000000) (rho,theta,phi)=(0.266569,90.000000,-154.927430)
TVector3 A 3D physics vector (x,y,z)=(-0.224983,-0.128987,0.000000) (rho,theta,phi)=(0.259336,90.000000,-150.173576)
. . .
```

Conclusions

□ TAGelement

- ➔ New class to handle MC and object relation

□ TAGdataStorageManager

- ➔ Class that handles relation between container objects
 - ➔ In development phase
 - ➔ No protection against wrong request (e.g.: try to retrieve TAVTcluster from TAITtrack)
-
- ➔ Once operational, replace TClonesArray by relation vector only
 - ➔ Retrieving object via data storage manager only