a proposal for discussion on how to proceed for muon cablings

+ a few considerations on muon converters

Stefania.Spagnolo, Dip. Fisica, Univ. del Salento and INFN Lecce

General strategy

- Implement review recommendations in MIG5 nightly build
- move bundles of tags into dev nightly builds (standard development release) when some coherent functionality is achieved

Impact on trigger

- very limited new coding on EF: algos depend on cablings only via rdotoprepdata tools
- maybe big on muFast (and LVL2 algos ?):
  - muFast directly depends on cablings, XXXgeometry, rawdataproviders tools;
  - performance in terms of cpu time is an issue at LVL2

Muon Cablings: a proposal for how to proceed

my own schema (to be negotiated/discussed/revised)

- 3 problems to be factorized
  - define package structures and basic (common to all technologies?) interfaces for translation between identifiers
    - very basic step ⇨ first thing to do
    - some interfaces/functionality will be new
  - access to maps in COOL, versioning with use of IoV service
    - can proceed in parallel (doesn't affect clients)
  - implementation of maps; revise transient model of cabling
    - keep (extend if necessary) existing model *as a start* ⇨ cpu time for accessing connectivity information does't change w.r.t. existing code ⇨ preserve LVL2 performance when muFast migrated to new cabl.svc
- requirement: don't break LVL2 (even in mig5) until a minimal functionality of the new structure is achieved
  - have a new structure of packages for cablings along with current MuonCablings in mig5; use new packages in offline only as a start

Muon Cablings: a proposal for how to proceed

my own schema (to be negotiated/discussed/revised)

- requirement: allow, also in the long term, muFast to use dedicated tools *(now XXXgeometry)*
  - for fast association of positions in space to online-identified hits
  - depending on (i.e. initialized with) cablings Svc.s and Geometry Svc.s *(use of Amdcsimrec not questioned at the moment)*
- create MDTgeometry package (move there the relevant class) along with RPC,TGC,CSCgeometry
  - maybe (change name) have all of them somewhere under trigger ?
- add MuonCablings/MuonXXXCabling
  - MuonMdtCabling: in the Svc, manage the access to the map and the arrays in memory representing the maps (now in MDTgeometry); define (+implement if possible) the extra methods needed (for example, provide rob id for a CSM id)
  - MuonRpc/TgcCabling: as before (i.e. skip the step of the singleton currently implementing the cabling model); extend interfaces (drop geometry based interfaces)
  - as soon as a minimal functionality is achieved XXXgeometry implemented via MuonXXXCabling

Muon Cablings: a proposal for how to proceed

my own schema (to be negotiated/discussed/revised)

- define/study requirements of LVL2: Alessandro's feedback extremely useful and welcome
- consider if a xml dictionary of online indices (as for the offline indices) to be parsed for the initialization of helpers connecting set of online indices to hash indices for rdo data collections
  - centralize hash functions management
  - leave to cabling svc to define the pure online to offline (and reverse function) connection
- prior to re-definition of interfaces: make an effort to summarize all needed sets of indices for all technologies
- study if there's any benefit to have a basic abstract interface for all technologies

Converters point 2 in the twiki

Tools (shared by online and offline)

RawDataProviderTools? `depend on RobDataProvider? + cablings + decoding tools`

RdoToPrdTools? `depends on cablings + decoding tools + RawDataProviderTools?`

Offline Algorithms

RawDataProvider? `use RawDataProviderTools? to convert bs for the entire event into RDO;`

`RawDataProviderTools? must have a dummy interface like convertAll (and internally it will know which are the ROB data fragments to request and decode)`

RdoToPrdAlgorithm? `use RdoToPrdTools? to convert the entire RDO of the event into PRD; RdoToPrdTools?` **`have alaredy now`** `a suitable interface to do that.`

Use in the EF (HLTalgorithms drive the data conversion by using the Tools [exactly same implementation as for the offline])

given a RoI, ask the region selectors what offline data collections (prd) are of interest: *i.e. get a vector of offline hashId*;

ask the RdoToPrdTools? for the pointers to the data collections corresponding to the list given by the Region Selector;

the RdoToPrdTools? (via the cablings) will determin which RDO collections (hash Ids) have to be accessed for the offline Id requested;

in the online (running on bs in the DAQ/HLT) the RdoToPrdTools? will request the pointers to the RDO collections of interest to the *RawDataProviderTools*;

the *RawDataProviderTools*, as usual, will be responsible for requesting the necessary ROB fragments to the RobDataProvider?;

in offline emulation of the EF (running on RDO), the RdoToPrdTools? will just read from SG the RDO collections of interest.

Converters – a small change needed for the EF functionality

in the EF we might request from TrigMuonEFSegmentFinder (for example) the conversion of the whole event in prd format (notice that currently in order to do that we actually run the offline algos before the trigger steering) by passing to the rdo->prd tools an empty vector of hash ids (like we do in the offline algorithms).

In the EF, however, it can happen that 2 lvl2 Roi in the same event seeds TrigMuonEFSegmentFinder. Therefore, I would like that, in case of convertion of the whole event, this is at least done only for the first roi. To do that I'd like to flag in the rdotoprd tools the situation of "first call in the event for an unseeded decoding" by introducing a bool data member taking a record of such situation.

```
if (!m_sgSvc->contains<CscStripPrepDataContainer>(m_outputCollectionLocation)) {
   /// clean up the PrepRawData container
   m_cscPrepDataContainer->cleanup();
  .....
    if (givenIDs.size() == 0) m_fullEventDone=true;
    else m_fullEventDone=false;

 } else {
   if (m_fullEventDone)
   {
      *m_log<<MSG::DEBUG<<"Whole event has already been decoded; nothing to do"<<endreq;
      return status;
   }
   if (givenIDs.size() == 0) m_fullEventDone = true;
}
```

any objection ?