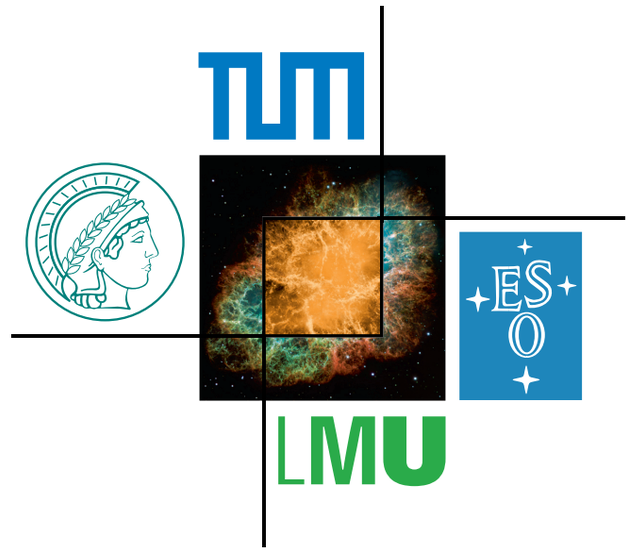
Case study:
Migrating callbacks from LArBadChannelTool

Jovan Mitrevski

June 6, 2016

Software TIM, Glasgow



Overview of current setup

- Disclaimer: I am neither a calo expert nor a conditions DB expert.
- I attempted to migrated the callbacks located in the LArBadChannelTool package, LArBadChanTool.{h|cxx}, to use the new framework.
- LArBadChanTool sets up two callbacks:

updateFromDB
reads from: CoolFolder (/LAR/BadChannels/BadChannels) or ComplimentaryCoolFolder (null)
fills: part of m_HwBadChan of type BadChanInfo (and m_State)

updateBadFebsFromDB
reads from: CoolMissingFEBsFolder (/LAR/BadChannels/MissingFEBs)
fills: part of m_HwBadChan of type BadChanInfo (and m_BadFebs)

- Also provides interface to access m_HwBadChan, allows writing (and overwriting based on text files), fills m_OfflineInfo based on m_HwBadChan and m_BadFebs. (Can it just use m_HwBadChan?)

Migration

- (Did not attempt to migrate full functionality.)
- Use CondInputLoader to put raw conditions into Detector Store

```
folderList = [("CondAttrListCollection", "/LAR/BadChannels/BadChannels"),
              ("AthenaAttributeList", "/LAR/BadChannels/MissingFEBs")
             ]
topSequence.CondInputLoader.Load += folderList
```

- Read these values using a conditions handle in LArBadChanAlg:

LArBadChanAlg

reads from:

BadChannels read cond handle

BadFEBs read cond handle

writes to:

LArBadChannels write cond handle

(type BadChanInfo, as m_HwBadChan)

Observations

- Quite straight-forward to do the basics, but subtleties need expert guidance:
  - Should there be both a BadChannels and ComplimentaryBadChannels read conditions handle?
  - How much text file input support should be provided.
  - How should offline bad channel info be presented?
    - Should it be another object of type OfflineInfo written out with a write conditions handle?
    - or should it be incorporated into the BadChanInfo data?
  - Alternately, the algorithm could be split, one just processing BadChannels (and potentially ComplimentaryBadChannels), the other BadFEBs.

Observations (cont.)

- The user interface needs to be thought out given the new scheme:
  - Data types currently used in the caches may need revising. Potentially they should be moved to new packages.
  - Should the data types themselves have an interface, or should something else, like a service, be used to access them?
    - Currently some things, like the FebHasher are the same for each object but stored in BadChanInfo. (The FebHasher also takes a pointer to the cablingService tool.)