

Trigger s/w plans

Tomasz Bold for trigger

UC Irvine, UST-AGH Krakow

Topics

- EDM
 - Sizes (status & plans)
 - Missing information (TrigPassBits)
- Bunch crossing tool
- Trigger Time Lines – understanding the trigger evolution

EDM

- Size of trigger EDM is bigger from release to release
 - Mainly because the menus grow (more triggers → more objects)
 - Overall 165 collections in ESD and 134 in AOD (irrespectively of the configuration)
- Two options to shrink down the size
 - Reduce number of triggers
 - This limits r&d on new triggers in physics groups
 - Prescale low p_T triggers in most of samples
 - Done in MC_pp_v1_tight_prescale → size down to ~130 kB/evt
 - Reduce number of collections in main productions (suggested during trigger Amsterdam workshop)
 - Will certainly limit possibilities of studies by trigger experts but maybe the studies of performance of new triggers in pys&perf groups will not be affected!
 - Your feedback needed here! I.e. we need to scan somehow all the analyzes and figure out what are the losses if we remove say: L2CaloClusters ...

EDM - what PAT wants - slimming

- Navigation slimming – this is implemented [see: HLTrigNavigationSlimming](#)
 - Not allowing to drop parts of collections (slim)
- What we always had on the todo list is collections slimming.
 - Though it is complex task → the top structure above trigger EDM is TrigNavigation build of TriggerElements. [HLTSteeringNavigation](#)
 - In it's persistent form it references objects in collections though indexes (begin,end) when collection is slimmed this indexes need to be recalculated taking into account the object sharing → that was believed to be very complicated and therefore not approached.
 - We are short of manpower here!

EDM – what PAT wants - uniformity

- Common base class. This is basically impossible!
 - Manpower.
 - Too much of diversity in trigger EDM to accomplish that.
 - Plus the experiment is running.
- What seems feasible is adapter i.e. to the I4Momentum (you need to tell us). Say Trig4Momentum to which the objects would be convertible.
 - So the matching framework can be implemented in terms of Trig4Momentum in most cases. Persistifying this needs probably some more thought.

EDM - TrigPassBits

- Sometimes in RoI there is more than one object → collection of objects → so the need to store decisions per object
- Implemented TrigPassBits (+helpers) to mark trigger feature as “passing” and then decode it back
 - Object is loosely mapped to the collection it is marking → therefore one type can serve Bphys, TrigElectrons, ...
 - In rel 16.1.X TrigDecisionTool is internally instructed to use this extra information once TrigPassBits are available → will be propagated to all releases
 - That should resolve few problems during analysis → there were already several posts on trigger-help HN
 - Note this are not only strange topological trigger, regular/plain electron & muon triggers will need this feature too. Will call trigger developers this week to update the codes so next year we resume running recording complete information
 - Object is optimized for the size (literally one bit per decision) – one collection for the whole trigger

TrigBunchCrossingTool

- Has a generic interface for inspecting the bunch structure of the data being analyzed
- Contains some largely independent implementations of the same interface:
 - TrigConfBunchCrossingTool: Reading the info from the COOL trigger configuration (source Frontier/in file metadata)
 - LHCBunchCrossingTool: Reading the info from the COOL TDAQ DCS folders
 - MCBunchCrossingTool: Usable on pile-up MC files, reading the bunch structure from the file metadata
 - WebBunchCrossingTool(SA): Reading the configuration from <http://atlas-trigconf.cern.ch>, usable outside Athena

Trigger Time Lines

- Which is lowest p_T unrescaled muon trigger in period E? For the whole period E was the trigger X the same? When I add period F should I use the same trigger or move to X_tight? I see that first few runs of period H the trigger X had apparently different turn on, what has changed?
- So far signature groups experts keep track of this
→ communication channel through wikis.
- The time has come to implement infrastructure to acquire and publish this sort of information uniformly.

Trigger Time Lines

- This will be extra database which will store this information.
 - Majority of the information will be distilled from existing databases. I.e. TriggerDB, COOL, releases.
 - Expert knowledge (now on wikis) should be easy to insert – this is important ingredient!
 - Plan is to make it accessible via web + command line to upload + integrate well with run-query GRL.
 - we seem to lack of manpower to do more on the web ... however it would certainly be useful
- More info: [Amsterdam core s/w talk](#)
[Joerg's initial slides](#) see also backup slides.

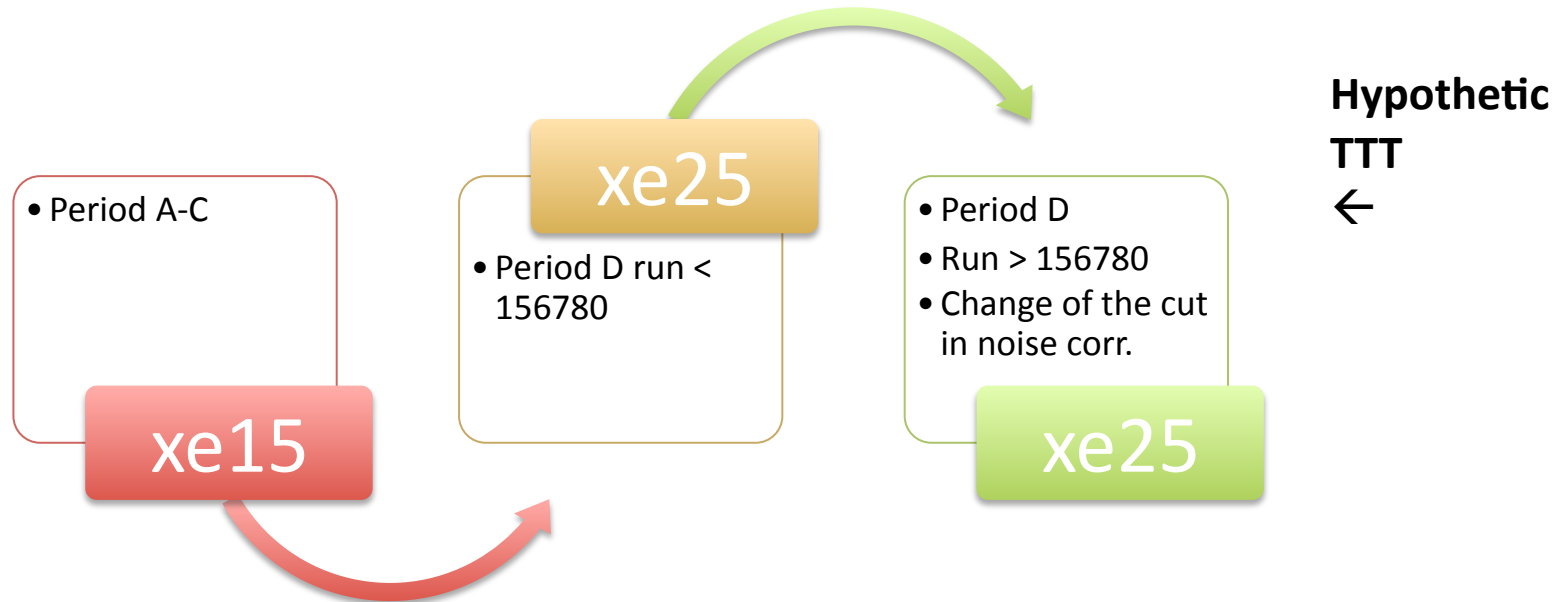
Backup

Triggers versioning

- Following triggers evolution is hard task
 - It depends on many aspects
 - Detector statuses, algorithms configurations & code, prescaling, conditions data, ...
 - So far this is traced
 - by data periods, trigger chain/items versions
 - And recorded in twikis and slides → go back to yestarday's talks
 - Assessment can only be made a posteriori
 - Needed in order to:
 - split data sample into stable trigger periods
 - jump from one trigger to another (if un-prescaled trigger is a must)
 - ...
- System somewhat similar to DQ flags is needed

Trigger Time Track

The timeline for lowest unrescaled MET chain used in $W \rightarrow \text{numu}$ analysis background



- Some of the breakpoints will be suggested (i.e. we know chains versions, releases)
- But expert knowledge will be needed
 - To drop irrelevant breakpoints
 - To introduce new
- The system is after 1st design proposal
- Your feedback is essential!