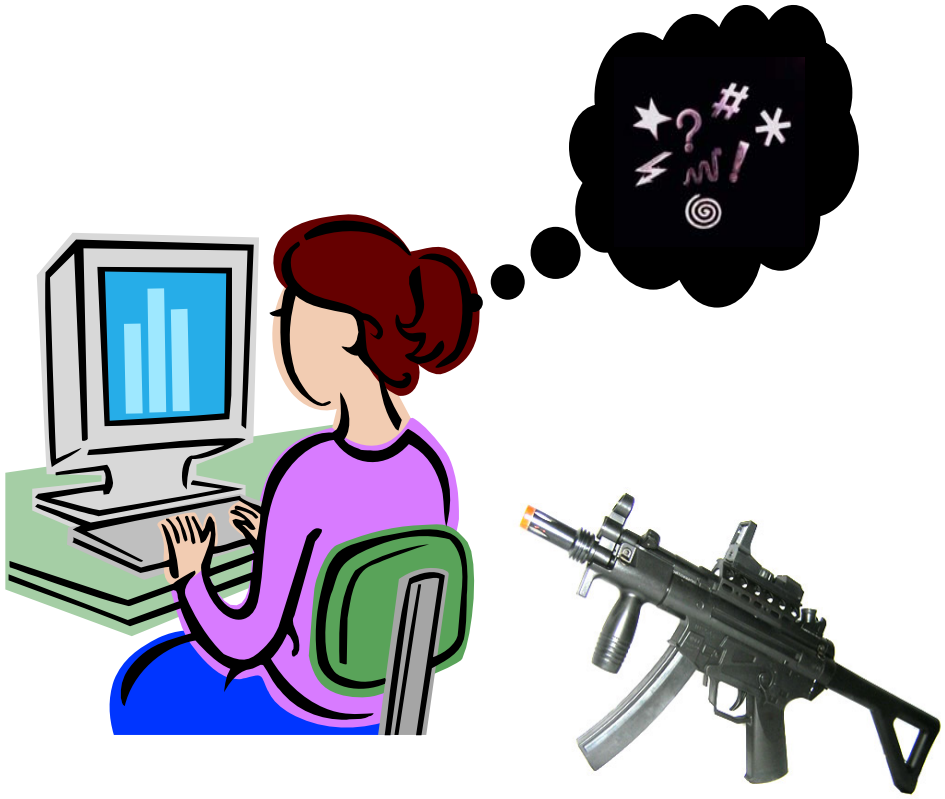
Trigger-aware analysis



- Current status
- Under development
- What's missing
- Conclusions & Outlook

Ricardo Gonçalo (RHUL)

Introduction

Amongst trigger objectives in CSC:

- Provide trigger simulation, supplying trigger information in AOD
- Support trigger-aware analysis
- Provide full trigger chain and configuration as a tool to develop realistic initial menu

For release 12 we want:

- LVL1+HLT trigger objects in ESD/AOD
  - Including trigger decision, data objects and navigation info
  - Support trigger-aware physics analysis: re-running trigger, truth association, etc…
  - Help develop and debug trigger menu and reconstruction
- Algorithms and menus for all slices

Current status

Using TriggerDecision

- TriggerDecision object now being filled for all signatures in 12.0.1

- Retrieve from StoreGate:
  - Default SG key is "`TriggerDecision`"
  - Currently modified by jobOptions to "`MyTriggerDecision`"

  (`TrigDecisionMaker/jobOfragment_TriggerDecisionMaker.py`)

- Query TriggerDecision:

```
TriggerDecision* p_trigger_decision;
sc = m_storeGate->retrieve(p_trigger_decision,"MyTriggerDecision");
if ( p_trigger_decision->isTriggerPassed() )
   std::cout << "Some signature was satisfied!" << std::endl;
if ( p_trigger_decision->isDefined("e25i") ) {
   bool e_candidate_found = p_td->isPassed("e25i");
   std::cout << "Some signature was satisfied!" << std::endl;
 }
```

- Filled for whatever signatures were running
  - Retrieves configuration tables at initialization for L1/L2/EF

Using TriggerDecision (cont.)

- Caveats:
  - Currently all intermediate steps also listed as signatures (but never accepted) due to feature of the Steering
    - will be corrected for the new Steering code
  - When re-running trigger, SG key gets added “+”:
    - “MyTriggerDecision+”, “MyTriggerDecision++”, etc (any better idea?)
  - TriggerDecisionMaker will need to be updated for new configuration (Trigger DB etc)

- “Invisible signatures” can be set:
  - Useful to have low-threshold signatures whose function it is to create data objects (tracks, clusters, vertices)
    - To allow real signatures to be be re-run on the ESD/AODs
  - Just add invisible signatures’ names to IgnoreItemNames list in jobOp

```
TriggerDecisionMaker          INFO Initializing TriggerDecisionMaker...
TriggerDecisionMaker          INFO Properties:
TriggerDecisionMaker          INFO  doL1             = True
TriggerDecisionMaker          INFO  doL2             = True
TriggerDecisionMaker          INFO  doEF             = True
TriggerDecisionMaker          INFO  TrigDecisionKey  = MyTriggerDecision
TriggerDecisionMaker          INFO  TrigConfigL2Key  = storeL2Location
TriggerDecisionMaker          INFO  TrigConfigEFKey  = storeEFLocation
TriggerDecisionMaker          INFO  IgnoreItemNames  = dummy0 dummy1
```

Persistency/Serialisation/AANtuple

- All trigger objects must be serializable into bytestream (through L2/EFResult)
  - No other way to save online information
- For offline running, mixed solution is to store objects in POOL and reference in L2/EFResult
  - This is necessary in some cases: e.g. to allow track-truth association
- “Physics-analysis” data should be POOL-only:
  - TriggerDecision will be turned into a Tool in real running: not a real data object in the long run
  - Track-truth matching only possible with Monte Carlo anyway
- Ntuple is also an issue:
  - Most of old CBNT infrastructure adapted to write AANtuples for easy offline analysis (ongoing work)
  - New data objects should also be written to AANt

Status* of trigger slices

✓ - ready or partially ready… usable anyway

✘ - non-existent or needs work

***as far as I could see: please correct me if I'm wrong**

	Level 1		Level 2			Event Filter		
Slice	ROI	POOL	FEX	Hypo	POOL	FEX	Hypo	POOL
B-tag	✓	✓	✓	✓	✓	✓	✓	✓
B-physics	✓	✓	✓	✓	✓	✓	✘	✓
Electrons	✓	✓	✓	✓	✓	✓	✓	✓
Photons	✓	✓	✓	✓	✓	✓	✘	✓
Etmiss	✓	✓	✘	✘	✘	✓	✘	✓
Jet	✓	✓	✓	✓	✓	✓	?	✓
Muons	✓	✓	✓	✓	✓	✓	✓	✓
Tau	✓	✓	✓	✓	✓	✓	?	✓

Additional functionality (not exhaustive):

- EF track-truth association is done (Iwona)
- TrigVertexCollection storable (Julie)
- AANtuple writing for L2 tracks and clusters (John)

Under development

Using offline analysis tools

- Ongoing discussion since Mainz
  - It was always at bottom of priority list because of lack of resources
- To make use of offline analysis tools (i.e. EventView) possible, trigger objects must inherit from:
  - INavigable4Momentum
  - NavigableTerminalNode
  - P4PtEtaPhiM (new 4-momentum base requested by trigger)
- Currently being tested in Tau slice for TrigTau (Olga, Carlos)
- Trigger requirement:
  - The new inheritance structure cannot affect persistification, serialization or time performance
- If TrigTau is successful do the same for other trigger data classes
  - In principle only applicable to classes which are 4-momentum like

Other things

Re-running level 1:

- Re-running CTP_decision was tried in the muon slice (Attila) but still early days
- What makes it hard is that hardware-trigger emulation has in principle to be re-run: can only ask for different multiplicity, but NOT change thresholds

Level 2 track-truth association:

- Requirements
  - Keep all possible matches
  - Keep mother-daughter relations within matching particles (bremstrahlung)
- Offline track-truth association not suitable for L2 (different track classes and requirements)
- First version in principle done and in 12.0.1, but persistification not possible (under investigation)

Complex menu:

- Several signatures exist for taus, b-tag, muons, photons and electrons up to level 2
- New functionality needed in Python configuration scripts (John)
  - Add multiple-object signatures: 2e15i, 2γ20i
  - Add multi-slice signatures: mu20+e15i, e25i+Etmiss
- Topological signatures on the way (See Till's talk)

Missing items

What's missing?

- Mapping between L1 RoIs and offline reconstructed particles (or at least existing object collections):
  - Was my offline electron candidate found by the trigger?
  - If not, where did it fail?
- Can we run many thresholds over each event? (optimisation)
  - Can this be done iteratively? (to approach optimum point instead of scanning whole parameter space)
- Documentation and analysis examples
  - Critical for success of trigger-aware analysis and to get feedback from physics groups!
  - Several "users" have started producing trigger-aware analysis. We must give them the right tools and examples.

Conclusions & Outlook

- Work progressing well to provide new functionality
  - Mostly in terms of having a baseline menu with many slices and signatures
- But we must keep our eyes on the goal: 12.0.3 must work seamlessly for trigger-aware analysis
  - In 12.0.3 need a reasonable baseline menu with all slices!
- Remember some (much) of this machinery will be the basis for bigger things:
  - Online debugging
  - Developing day-1 menu
  - Tools to help menu optimisation and evolution

Trigger-happy analysis