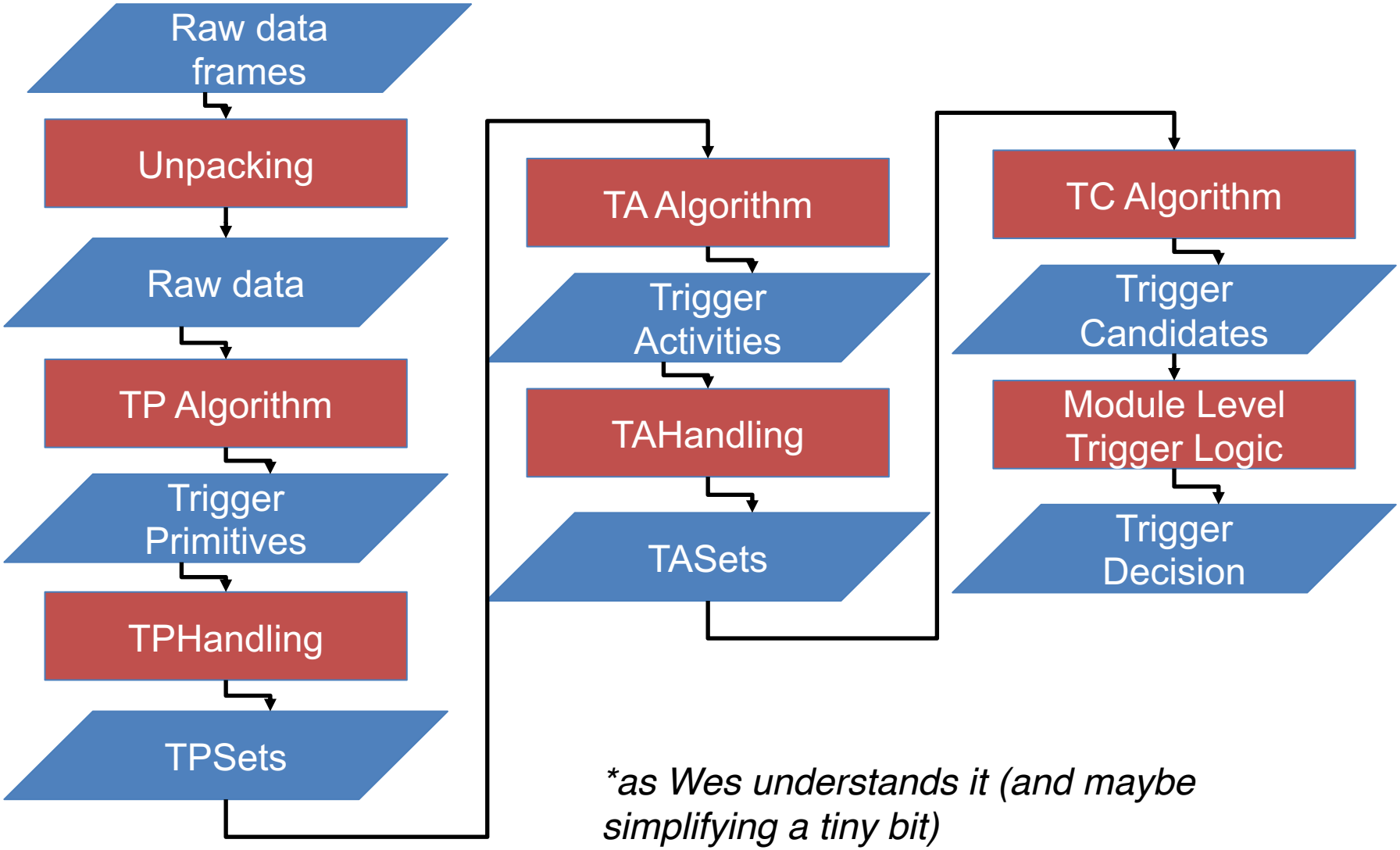
Trigger Chain Simulation in LArSoft: Update

Wesley Ketchum

DUNE DAQ DS / PP Meeting

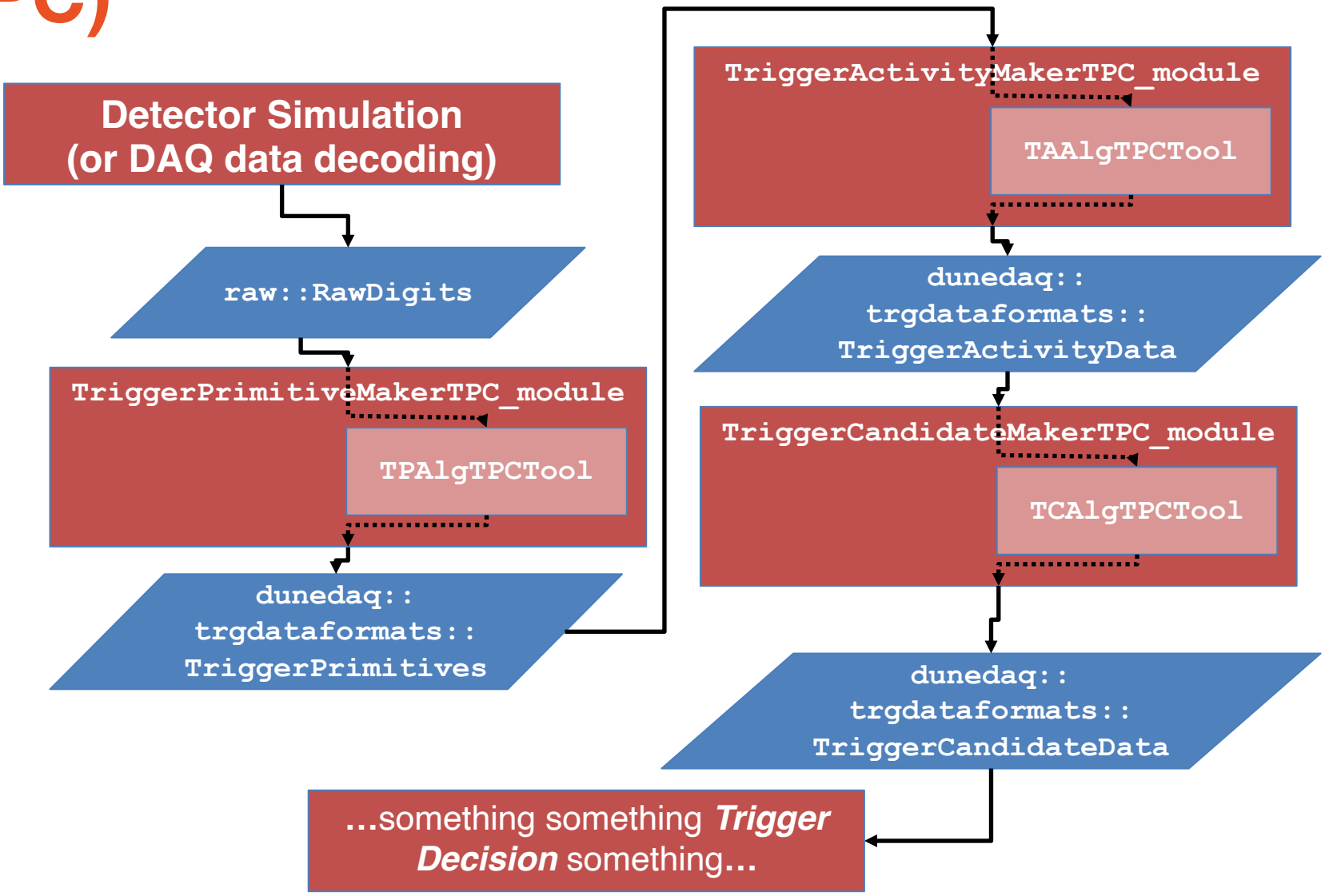
28 May 2024

Recap of general flow in DAQ*



as Wes understands it (and maybe simplifying a tiny bit)

Recap of proposed flow in LArSoft (TPC)



General philosophy / goals

- Use the same data types in online and offline
  - Create TPs, TAs, and TCs in the offline chain that can be directly compared to or used interchangeably with “online” created ones
- Use algorithm interfaces that are realistic for what is done online
  - E.g. TP algs run per channel, TA algs see one TP at a time, etc.
    - Ideally we could share algorithms across online and offline more directly…
- Decouple “data preparation” and the algorithms
  - Provide modules that will …
    - read the input data collections
    - organize data such that it appears like it would in online
    - write the output collections

Status

- A small group of us have been pushing on this over the past ~month or so
  - Michal, Hamza, Alex, Animesh, Alex, Ivana, Emanuele, …
- Code currently lives in a new `dunetrigger` repository
  - Personal github for the moment
  - Up to date with dunesw v09_89_01d01
- Recent updates
  - Addition of TriggerCandidateMakerTPC module (Michal)
  - Analyzers for beginning to perform comparisons across collections (Hamza)
  - Added associations of TPs to TAs (Wes)
  - First attempt at TP SimpleThreshold algorithm (Wes, Alex)

Some auxiliary decoding work

- Initial decoders of trigger objects exist in duneprototypes
- Needs some updates:
  - TAs and TCs not unpacked properly as they don't use the 'overlay' classes
    - Reminder: TAs are TriggerActivityData + array of contributing TPs, TCs are TCData + array of contributing TADatas
  - Doesn't store contributing objects for TAs and TCs
    - Now producing new "inTAs" collection of TPs, and the association of TAs to those TPs (and similar for TCs)
    - **Not** associating to TPs from readout collection: slightly harder to do and not guaranteed to be there
  - Currently stored as map of <SourceID, TriggerObject>, but don't believe we want to persist SourceIDs to offline (right?)
    - Another issue where we are storing TP data from both readout and trigger buffers (as well as TAs) → proposal is to remove buffer from trigger
- Updates on PR here, but needs some additional conversation on how to handle new versions of data

TriggerPrimitives

- TriggerPrimitiveMakerTPC_module
  - Reads in the RawDigit collection (raw waveforms)
  - For data, gets associated timestamp for each waveform
    - ‘default’ timestamp used for simulation
  - For each channel, runs a process_waveform function from a TPAlgTool to generate TPs

```
//take in a waveform, add trigger primitives to it
virtual void process_waveform(std::vector<short> const& adcs,
                              dunedaq::trgdataformats::channel_t const channel,
                              dunedaq::trgdataformats::detid_t const detid,
                              dunedaq::trgdataformats::timestamp_t const start_time,
                              std::vector<dunedaq::trgdataformats::TriggerPrimitive> & tps_out) = 0;
```

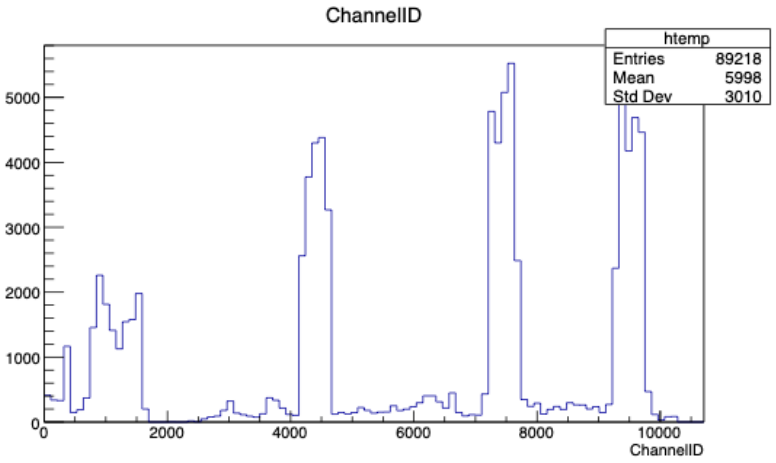
TPAlgTPC

- The TPAlgTPCTool allows for having a configurable algorithm selected within the module
- ***TPAlgTPC development should ideally be confined strictly to new TPAlgTPCTools***
  - Any change to the module or TPAlgTPCTool interface needs to be reviewed by trigger experts to make sure it is sensible
    - No doubt there will need to still be changes → we should try to 'get this (mostly) right' before making this more public
- TPCAlgTPCSimpleThreshold example now ready for testing
  - Intializes channel state for each channel at beginning of processing
    - Determine threshold to use based on plane
    - Pedestal set to the mode (most common value) of the entire waveform
  - Process tick-by-tick like the online does
    - Update pedestal determination, track going above and back below threshold, and create TP when below threshold

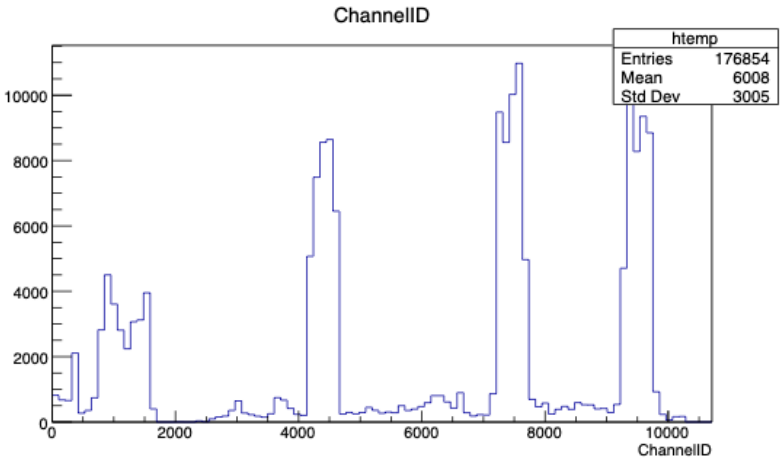
Offline / online comparison

- Hamza had an initial look (thanks Hamza!)
  - Comparison of offline implementation in LArSoft with data from last Monday (SimpleThreshold TPs at threshold of 500 ADC)
  - Online has twice the TPs as offline due to the duplicated TPs from the trigger's TP buffer
    - Looking to remove that and do more detailed comparisons

LArSoft TPCSimpleThresholdAlg



Online TPs (SimpleThreshold Alg)



TriggerActivities

- TriggerActivityMaker_TPC_module
  - Read in a TP collection
  - Sort TPs by ReadoutPlane (ROP) and in time
  - Per ROP, (re)initialize TAAlgTPC and call `process_tp`
  - Write ***TAs and associated TPs*** to the event
- TAAlgTPCTool for configurable algorithm selection
  - Again, ideally, developers only create new TPAlgTPCTools following this interface
  - Don't yet have a realistic algorithm in place here

```
class TAAlgTPCTool {

public:

  typedef std::pair< dunedaq::trgdataformats::TriggerActivityData,
                     art::PtrVector<dunedaq::trgdataformats::TriggerPrimitive> > TriggerActivity;

  virtual ~TAAlgTPCTool() noexcept = default;

  virtual void initialize() {};
  virtual void process_tp(art::Ptr<dunedaq::trgdataformats::TriggerPrimitive> tp,
                                      std::vector<TriggerActivity> & tas_out) = 0;
};
```

TriggerCandidates

- TriggerCandidateMakerTPC_module
  - Following from the example from the TriggerActivityMakerTPC
    - Sort TAs by time before
    - Thanks Michal for spearheading this!
  - Still needs the associations of TriggerActivityData added
  - TCAlgTPCTool follows a similar model to everything else as well

Running the chain

- Simple fcl configurations for running TP chain
  - Example at right for TP and TA algs
  - Note: separation of module configuration from tp/taalg configuration

```
producers:
{
    tpmakerTPC:
    {
        module_type: TriggerPrimitiveMakerTPC
        rawdigit_tag: "tpcrawdecoder:daq"
        tpalg: {
            tool_type: TPAlgTPCSimpleThreshold
            threshold_tpg_plane0: 100
            threshold_tpg_plane1: 100
            threshold_tpg_plane2: 100
            verbosity: 5
        }
        verbosity: 1
    }

    tamakerTPC:
    {
        module_type: TriggerActivityMakerTPC
        tp_tag: "tpmakerTPC"
        taalg: {
            tool_type: TAAlgTPCExample
            multiplicity: 100
            verbosity: 1
        }
        verbosity: 1
    }

}



stream1:  [ out1 ]
reco: [ tpmakerTPC, tamakerTPC ]
trigger_paths: [ reco ]

end_paths:     [stream1]
}
```

Ready to push to offline?

- Not yet (I think); should …
  - Finish validating LArSoft algorithm, and give reasonable thresholds for offline simulation
    - May not be perfect, but we want to provide realistic TPs for offline studies, especially for the far detectors
    - Implement StandardRunningSum?
  - Implement more realistic (even if still simple) TA and TC algorithms
    - ADCSimpleWindow TA and TC algorithms likely good
    - Here, mostly to provide a reasonable example of TA and TC algs, but would be ideal if we validate behavior against online implementation
    - Should also to the TC associations to TA data
  - Validate running chain on far detector simulation and produce sample fcl configs for adding to production
    - Note: production mode where we have access to RawDigits will be required