Simulation for Level 2

James T. Linnemann
Michigan State University
NIU Triggering Workshop
October 17, 1997

Simulator Goals

- Detailed simulation: *Experts*
  - develop trigger algorithm
  - verify trigger behavior (online v.s. simulation)
  - debug nasty events
- Physics simulation tool *Experts and Users*
  - evaluate efficiencies, backgrounds
  - studies to set up triggers

Simulation Requirements

- Serve Users and Experts
  - Users = no new code, so no relink
  - Users more interested in multiple platforms
    - at cost of less precise simulation?
- Run on real (C++?) or on MC (Zebra?) data
  - which level? Raw, STA, DST, μ DST
    - "natural" interface is raw (L1)
    - not always available or fully recoverable
- Possible to drive from real trigger scripts
  - not easy for non-experts
  - many studies can be done with object ntuples

User accessibility

- Object ntuple outputs (*same for any script*?)
  - tag with quality flags to cut on
  - what level object?
    - L1 objects ill-defined? “a combination passed > 10GeV”
    - L2 objects in preprocessors, or after cross-detector matching in global?
- How coupled are L1, L2, L3 simulations?
- Auxiliary output when trigger scripts needed?
  - Overlaps between bits?
  - Object traceability to what precision?
    - Tag objects with L1, L2, L3 bits passed?
    - further tags to parameter sets (in run header)?

Trigger Objects w/o Scripts? Design Trigger to Ease Simulator?

- L2 preprocessors with 1 variable parameter?
  - Just lowest Pt object to save?
  - But some quality cuts are in preprocessors
  - Can't do once & for all if algorithm or cut vary
- L2 Global Objects:
  - here finally match across detectors
    - or even with L1 objects--depends on trigger script?
  - separate out matched object lists?
    - Online code might want to stop asap?
    - Online code might not run all matches every event?
    - Match code buried in higher level tools?

Code Releases

- *Must* have Production release, version stamp
  - on MC too
  - couple simulation to online trigger releases?
- What is a package?
  - L1, L2, L3?
    - Or Lower level processors, and frames/hi-level tools?
- Avoid coupling with RECO, GEANT? How?
- Constants, trigger hardware not fully captured
  - Database had no “releases”

(Extra) Design work needed

- L1: Programmer interface
  - code to represent L1 HWFW to combine terms
    - L1 FW is hardware
    - L2, L3 it's software--code simulates itself so no new work
  - interface between term-makers and L1 FW
- L1, L2, L3: mostly user interface
  - object ntupler
    - like online monitoring histo's
  - bit-by-bit summary
    - detail levels from overall to per-bit to details (experts)
  - (getting trigparse script to drive simulator)

Scripts: Coor Programming

- Trigparse: defaults to squash compexity
- Interesting issue (Shared with L3)
  - “quality” = {high,medium,low}
    - defined by sets of cuts for each tool
    - enforce by hiding these cuts (tools really have just a quality setting)?
    - Or, trigparse substitutes these cuts for “quality” flag?