Level-3 Filter Real-Time Interface

– or –

it's been almost seven months since our last review, so why not?

Sarah Demers, Kevin McFarland, Tony Vaiciulis
University of Rochester

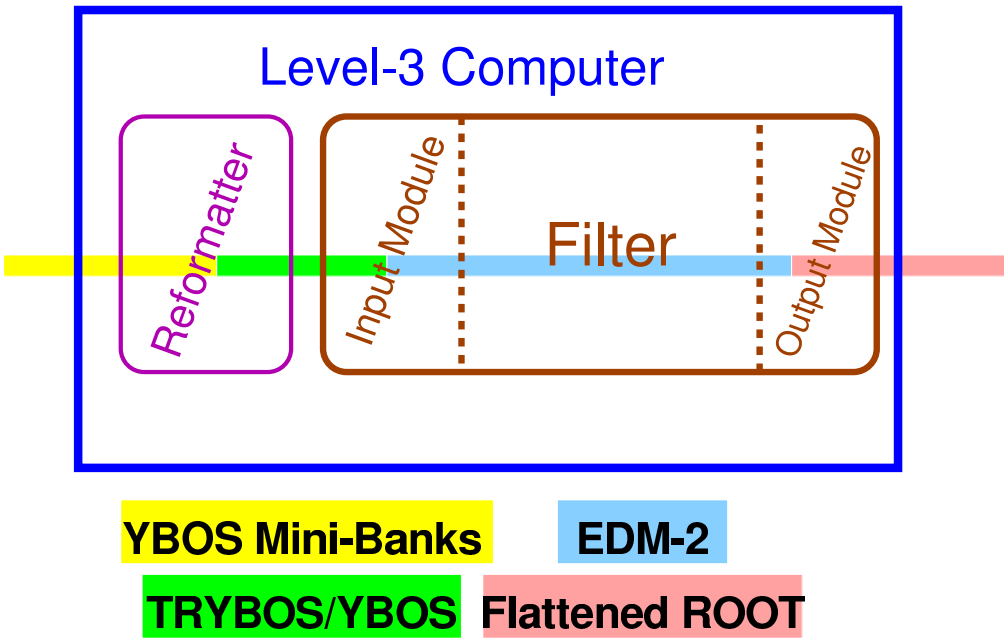
4 August, 2000

Outline

1. Online Functionality
2. Testing Procedures
3. Concerns and Future Development Plans

Event Manipulation

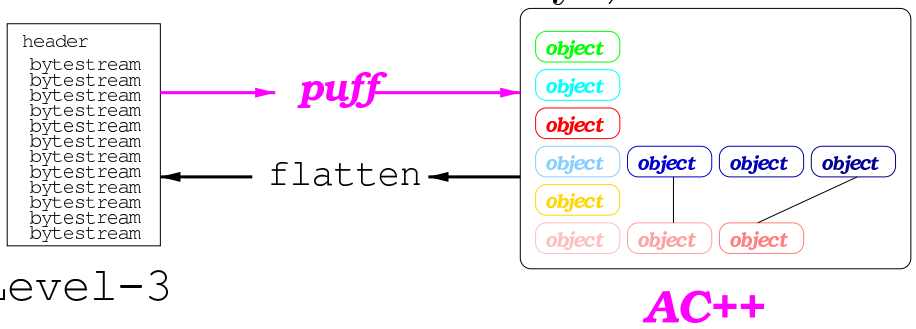
- Formats



▷ Event appears from reformatter in TRYBOS format

▷ Analysis must proceed on EDM-2 event (ROOT-based memory model)

▷ The EDM-2 event is "fluffy", but Level-3 sends BLOBs



▷ ROOT-Team has split "file-writing" operation into two steps: "fluffy" to "flat" and "flat" into file

▷ Ergo, two conversions required

Event Manipulation (cont'd)

- Event Content
  - ▷ Information in Header Banks
    - → Run Section (hope to use reformatter)
    - → Output streams
  - ▷ Trigger results and monitoring information
    - → TL3D (trigger and stream results; filter module results)
    - → TL3B ("TAGC" replacement), TL3T (time per module)
- Online (Temporary) Information
  - ▷ Decision (think: Roman emperor)
  - ▷ Data hub routing and file-writing information in a temporary header (discarded at CSL)

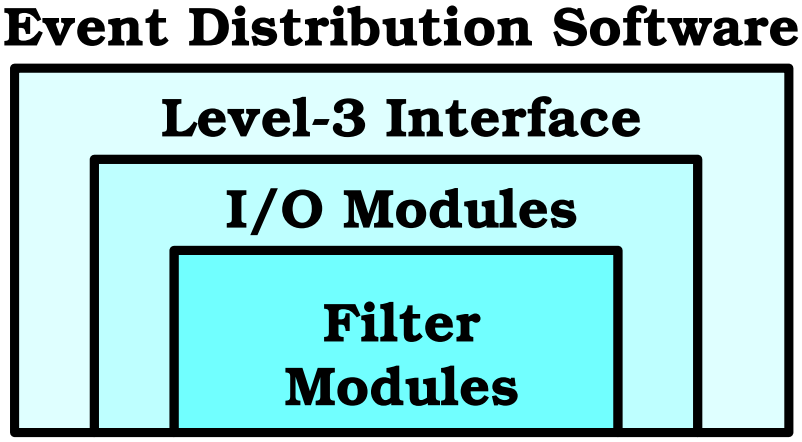
first word	length (words)	description	bitstream?
0	1	stream mask	√
1	2	Level 1 trigger mask	√
3	4	Level 2 trigger mask	√
7	8	Level 3 trigger mask	√
15	1	data representation	√
16	1	Level 3 executable number	
17	1	partition number	
18	1	run number	
19	1	run type	
20	1	trigger number	
21	1	crossing trigger number	
22	1	run (sub)section number	
23	9	reserved for future use	

AC++ Interface

- BOR/EOR
  - ▷ Mapping of DAQ messages to AC++ dispatch and filter completion
- Formation of Triggers
  - ▷ Triggers formed from AC++ paths

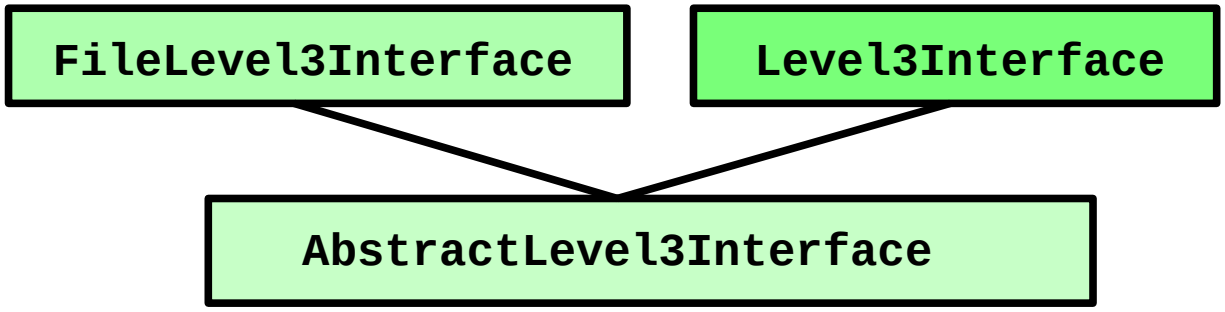
Testing Infrastructure

- Two Layered Design



  - ▷ Interface communicates between event-passing software and Level-3 Input/Output Modules
  - ▷ Level-3 Input/Output Modules handle event and `AC++` interface

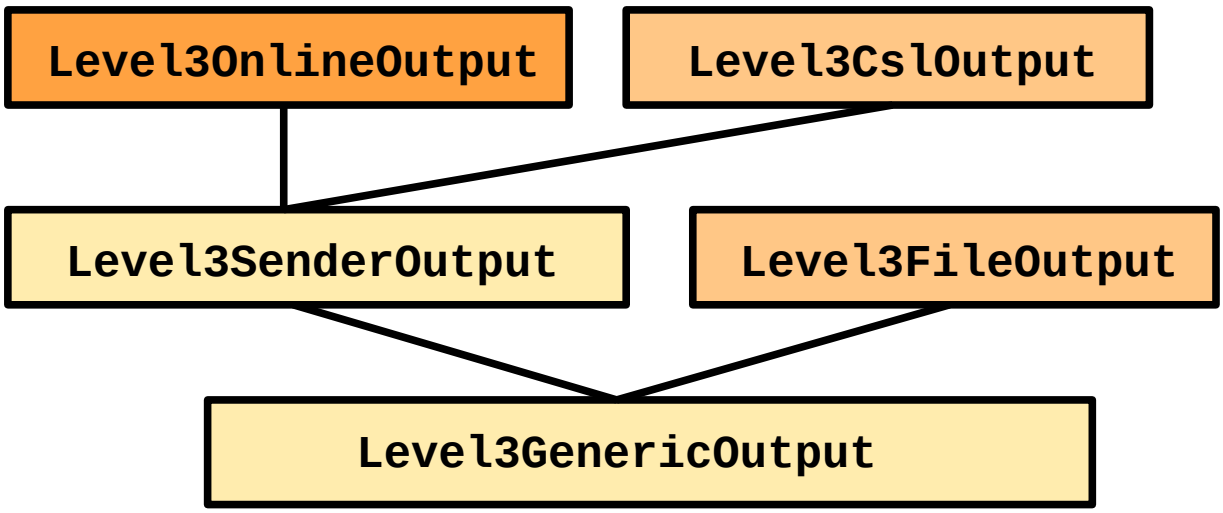
- Interface Classes



  - ▷ `Level3Interface` handles communication with the real Level-3 system
  - ▷ `FileLevel3Interface` emulates this, taking input from TRYBOS files and sending output to ROOT files
  - ▷ Selectable at run time!
    Easy to test actual Level-3 exe

Testing Infrastructure (cont'd)

- I/O Modules
  - ▷ `Level3Input` communicates with interface, converts event
  - ▷ `Level3*Output` adds event content, flattens



→ `CSL` flavor sends event directly to Data Hub (MDC-1 "Level-3 Lite" operation)

→ `File` flavor bypasses flattening operation

→ Only `Online` flavor uses interface

  - ▷ Test I/O Modules can be built into executable and run-time selected
- Testing Suites available for...
  - ▷ Level-3-CSL communication
  - ▷ Level-3 event content
  - ▷ Level-3 event conversion
  - ▷ Level-3 filters

Additional Work Needed & Concerns

- Need to mark "error" events for special handling (dedicated streams, special messages?)
- Need more robust event conversion on input (corrupt TRYBOS record will usually crash filter)
- New run control information added manually to interface; better to make a copy of configuration struct available(?)

Summary and Conclusions

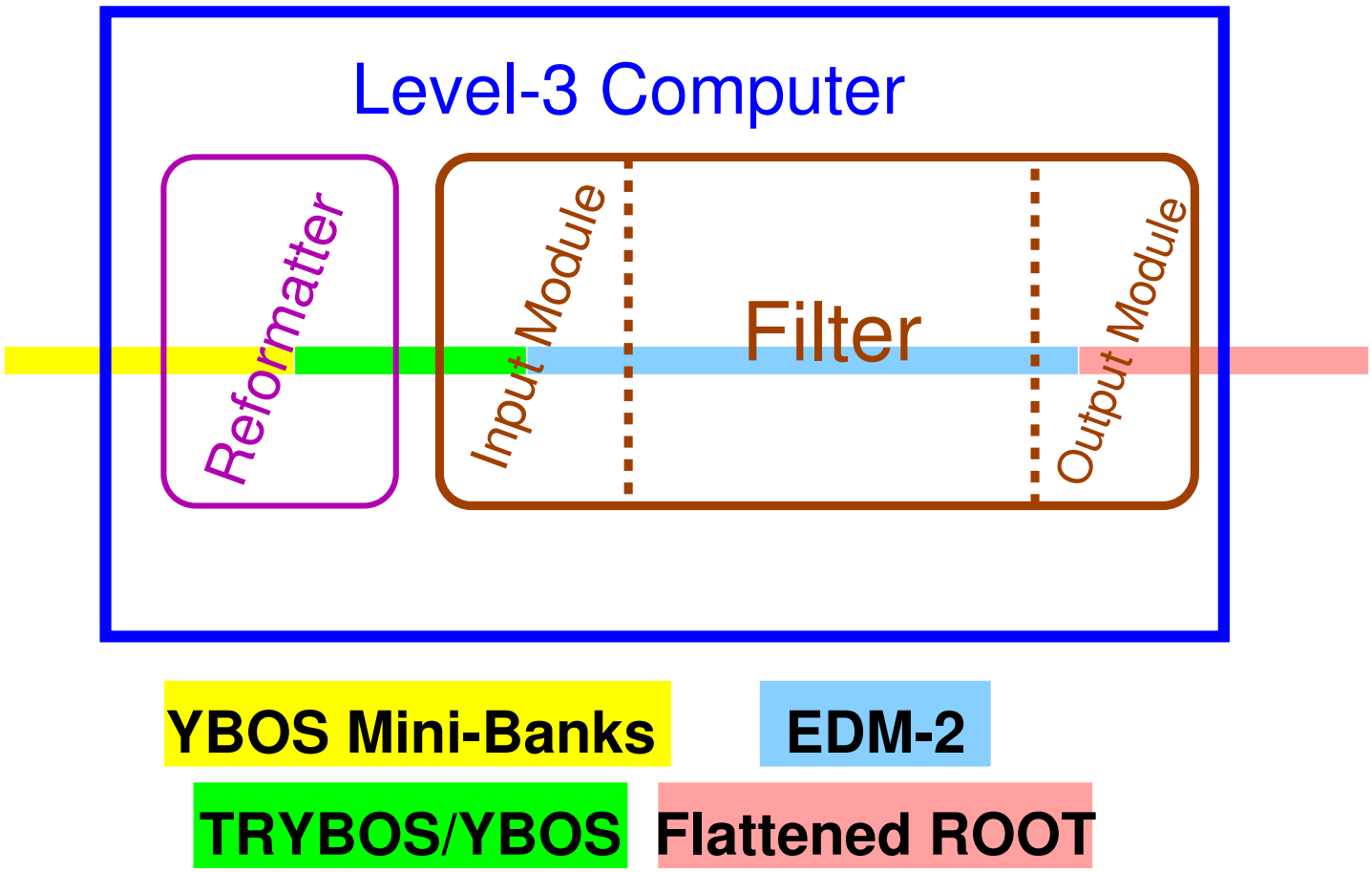
- Level-3 I/O infrastructure in place
- Significant test infrastructure also available (and in use)
- Some concerns: specifically error handling

A Word about Event Formats

Why EDM-2 in Level-3?

- Run II Computing God-Parent Committee recommends ROOT file format as compromise with event database proposals (December '97)
- To make ROOT format useful, post-production results in EDM-2 "objects" (generalized banks, supporting non-trivial links)
- Analysis code begins to be developed assuming these "objects" can be stored for inter-module communication
- In order to reuse production analysis code in Level-3 *(long-time stated goal of Level-3 project; c.f., D0)*, Level-3 must also adopt event model supporting objects
- **Completely predictable outcome for 2+ years**

Mechanics of Format Conversion



- Data assembled from VRBs in YBOS-like mini-banks
- Reformatter produces valid TRYBOS/YBOS event (flat)
- Filter interprets TRYBOS event, converts banks to Storable (EDM-2) Banks in "puffing" event
- Analysis code uses EDM-2
- Output of event is "flattened" ROOT, similar to the ROOT sequential file format
  (Functionality to back-convert Storable Banks – raw data – to Banks and write as TRYBOS being preserved)