

**Gamma-ray Large
Area Space
Telescope**



GLAST Large Area Telescope

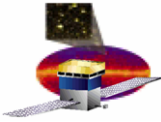
**Instrument Flight Software
EM2 Review
26 February 2004**

Event Delivery

**J.J. Russell
Stanford Linear Accelerator Center**

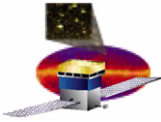
russell@slac.stanford.edu

650-926-2583



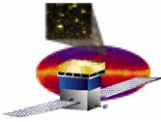
Event Delivery: Requirements

- **Flight Software General Requirements**
 - **Req. 5.2.1.3**: The EPU FSW shall receive fully assembled events from the Event Builder, formatted according to the custom hardware and software protocols defined in [5]. The event data shall be placed directly in the EPU memory.
 - **Req. 5.2.2.3**: The EPU FSW shall monitor event data for integrity and to track changes in event and detector statistics. The EPU FSW shall notify the SIU via the CPU-to-CPU protocol in the event of an error or anomaly.
 - **Req. 5.3.1.4**: The LAT science data interface shall accommodate data transfer rates to SC storage, up to a maximum of 66 Mbps. The SIU FSW shall format data into CCSDS telemetry packets, tagged with application IDs (APIDs) in the same manner as the CTDB telemetry, as specified in [7]. Each CCSDS source packet shall be preceded with a synchronization word (format TBD) to aid in data recovery, should synchronization be lost.



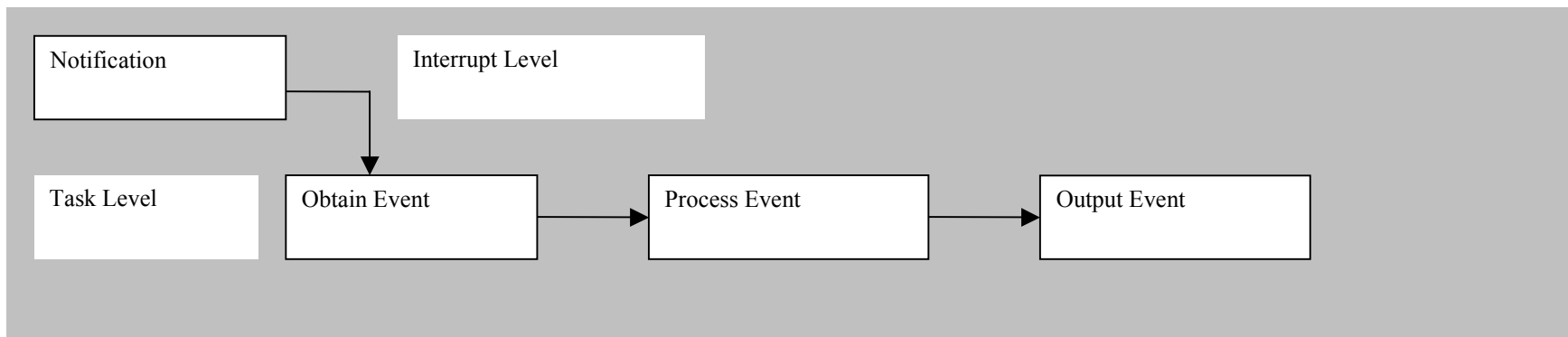
Event Delivery: Functional Components

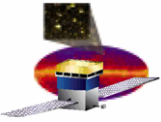
- **Functional Inputs from the Event Builder Modules**
 - The Event Delivery function receives interrupt notification via the LAT Communications Board when data is available
 - Once notified, the Event Delivery function begins to receive the asynchronous event data streams via the LAT Communications Board
- **Functional Processing in FSW**
 - The Event Delivery function routes event data for processing of various kinds. Event data can be simply passed through to output, be checked for integrity, or be filtered, compressed, etc.
 - The Event Delivery function delivers enough information for user code to reassemble entire events.
- **Functional Outputs to Spacecraft/ISIS**
 - Event data packets are delivered as a synch word plus the usual CCSDS packet to the Solid State Recorder on the Spacecraft
 - Science data for GRB detection



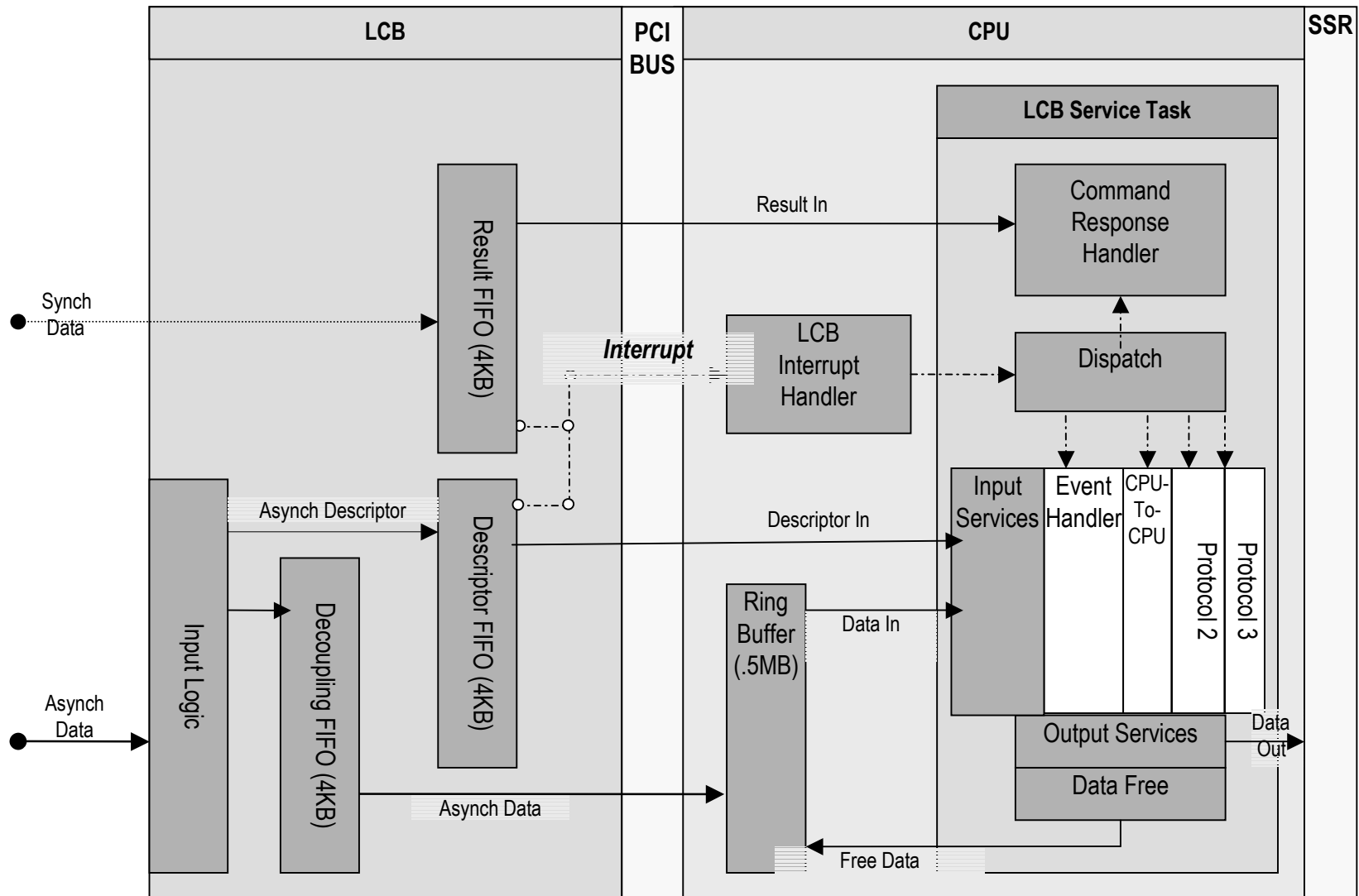
Event Delivery: Overview

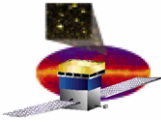
- One can think of the Event Delivery function as simply moving data through a pipeline, with hardware at the input valve and different hardware at the output valve.
 - Delivery of events is strongly hardware driven: the LAT Communications Board operates on both the input (LAT) side and the output (Solid State Recorder) side. The Event Delivery function is mainly concerned with efficient use of the LCB.
 - The Event Delivery function tries not to concern itself with controlling the input of event data; if events come in, they are processed.
- Event data moves through the pipeline in three primary stages:
 - 1. Notification
 - 2. Processing
 - 3. Disposition





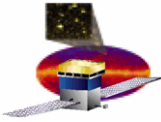
Event Delivery FSW: System Architecture





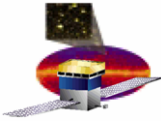
Event Delivery: Event Notification and Data Servicing

- The LCB Interrupt Handler is served an interrupt:
 - Whenever data arrives in the synchronous FIFO
 - Whenever a high-water mark of asynchronous data builds up in the ring buffer
 - The Interrupt Handler then disables the interrupt
- The LCB Service then determines which FIFO, the synchronous results FIFO or the asynchronous descriptor FIFO, has data. It does this by trying to remove an item from each of the FIFOs in turn.
- If the response FIFO is non-empty, the user-supplied response handling routine shall be called back to handle the data.
- If the descriptor FIFO is non-empty, one of four (one for each LATp protocol type) user-supplied routines is called back.
 - Because the data for all four protocols is piled up in the common ring buffer, the software has no choice but to retrieve the data descriptors in order. This in-order paradigm applies also to returning the data to the ring-buffer.
- The LCB Service Task shall continue to read both FIFOs until they are empty. When both FIFOs are empty, the LCB service routine shall re-enable the interrupts and wait until another interrupt is received.



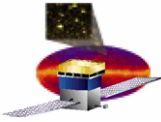
Event Delivery: Event Processing

- The Event Delivery function performs no “processing” of events. It routes event data descriptors to other functions for separate processing.
- Event Description
 - The LCB service task shall pass a descriptor of the event, containing its length and a pointer to its location in the ring buffer, to any event processing routine.
 - The event processing routine shall use this information to locate and unpack the event for further processing.
- Event Fate
 - The event processing stage shall determine the fate of event.
 - The event processing can dispose of an event in any one or all three of the following ways
 - 1. Return the event to the ring-buffer
 - 2. Pass the event to an output stage
 - 3. Transform the information
 - The event processor is also free to output more than one representation of the event. For example, the filter will, for a subset of events
 - 1. Reject the original event with no output
 - 2. For accepted events, compress the event data and queue it to be written to the SSR
 - 3. For gamma candidates, compose a small data block to pass on to the science analysis software.



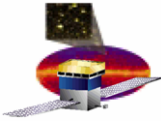
Event Delivery: Event Disposition

- There are two primary aspects to event disposition:
 - 1. Freeing the memory associated with the input event
 - 2. Writing the output event or some representation of it to an output device(s)
- Event Output
 - The Event Output stage shall be decoupled from the event processing stage if this stage would introduce unacceptable latency.
 - Because of the serial processing nature of the ring buffer, the processing of one event directly impacts the latency of
 - 1. Processing the next event
 - 2. Processing the next response descriptor
 - Therefore, if the writing of an event to the output device causes unacceptable latencies, the event will be placed on a queue to a separate task. Figures of merit for acceptable and unacceptable are on the order of 50 μ secs.
- Event Freeing
 - Either the Event Processing stage or the Event Output stage shall return the event to the ring buffer
 - Provided as an LCB Service



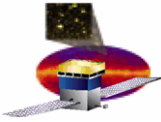
Event Delivery: Pipeline Synchronization

- In some cases, flow of data through the delivery pipeline may need to be synchronized. For instance, users may need to tell EPUs to start taking data at the same time.
- Simply sending a message to both EPUs from the SIU is insufficient. There is no guarantee when each EPU will service the message with respect to the incoming event stream.
- The only way to get both EPUs to react simultaneously is to broadcast something into the event stream using the SIU.



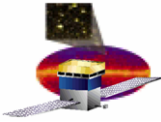
Event Delivery: Testing

- **Input Side Testing (data integrity)**
 - **Event Export / Import Loop-Back Data Integrity Test**
 - Using a loop-back cable, event packets with a known structure are transmitted to the input of the LCB and are then imported as asynchronous event data, which is then verified.
 - **Event Circular Buffer Wrap Around Test**
 - Verifies that ring buffer data is DMA'd correctly
 - **Event Descriptor FIFO High Water Mark Interrupt Test**
 - The high-water interrupt marker is set to each of its four possible values. Using the loop-back capability, precisely the amount of data needed to tickle the interrupt is then forced into the front-end, and the operation of the interrupt is tested.
 - **Ring Buffer FIFO High Water Mark Interrupt Test**
 - As above.
- **Output Side Testing**
 - **Data Integrity And Packaging**
 - Sample input events are placed in EPU memory, then written to the SSR, where they are checked for correct CCSDS headers.
 - **Performance Monitoring**
 - As above, but bandwidth to the SSR is tested.



Event Delivery: Testing (cont'd)

- **Complete Path Testing**
 - In these tests, the event source is switched from being the loop-back input to using the FES. The same set of tests (described above) are repeated, but the emphasis is on end-to-end performance.
- **Performance Testing**
 - The FES is set to deliver a known collection of events from all data sources at rates of 100Hz, 1KHz, 5KHz, 10KHz, 20KHz. A dummy processing stage is inserted that counts the number of events it sees. The number of received events must match the total number sent of events as measured by the GEM statistics counter.
- **Latency Testing**
 - Again, the FES will be used to inject events into the system. The arrival time of each event will be compared to the timestamp applied by the GEM. A histogram giving the distribution of the differences between this time and the actual time represents the event latency. This test will be repeated with the actual filter code, measuring the arrival time and the exit time of the event.



Event Delivery: Consumers

	Test Stands		ISIS	I&T	FSW Test-bed	FSW Deliverable
	TKR,CAL	ACD				
Real event delivery (instrument to CPU)	Y	Y		Y	Y	Y

- **Status:**
 - **LCB instrument and service task is complete.**
- **Test Stands:**
 - **Test Stand engineers require this function to evaluate whether their builds of the LAT are delivering data.**
- **I&T:**
 - **I&T requires this function to evaluate any fully-built version of the instrument.**
- **FSW Test-bed:**
 - **Development of instrument physics functions require the availability of a stream of event data.**