Status of VELO Software

Eduardo Rodrigues
On behalf of the VELO software group

LHCb Software Week, CERN, 10 December 2008

- Organisation issues
- Status
Organisation issues
Software commissioning

- Algorithms delivered … but software commissioning took 2nd priority compared to hardware commissioning

- Shutdown period being exploited for commissioning of the VELO software

- Workshop on 23rd Oct. to:
  - review status of software
  - identify critical items
  - dress a work-plan with list of milestones to achieve
Areas of work

- PVSS, DAQ recipes
- Timing and gain
- Error banks
- Vetra project
- Monitoring
- TELL1 emulation
- Alignment
- Closing software

Milestones defined for each area of activity, with priorities set and a responsible person attached

- Twiki page
  “Software milestones for 2008/2009 shutdown”
  https://lbtwiki.cern.ch/bin/view/VELO/SoftwareMilestones
Meetings & Co.

**Commissioning meeting**
- Mondays, software section at 10h15
- Report on previous milestones achieved
- “Round-table” news
- Plans for the week

**Integration days**
- Thursdays
- Used for integration of software releases, software tests at the pit, etc.
- Milestones Twiki page: [https://lbtwiki.cern.ch/bin/view/VELO/SoftwareMilestones](https://lbtwiki.cern.ch/bin/view/VELO/SoftwareMilestones)

**Weekly VELO meeting**
- Progress reports avoiding details
- Specific presentations on items of general interest
### Milestone Summary

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>29/10</td>
<td>PVSS restructuring, Test of naming panel and diagram</td>
</tr>
<tr>
<td>46</td>
<td>7/11</td>
<td>Parameter upload Plan, v6r3 firmware emulation release - deadline moved! (new Gaudi release date has been changed)</td>
</tr>
<tr>
<td>46</td>
<td>13/11</td>
<td>PVSS backup, Delay Scans</td>
</tr>
<tr>
<td>47</td>
<td>20/11</td>
<td>(Global Com.), MCMS panel</td>
</tr>
<tr>
<td>48</td>
<td>27/11</td>
<td>Firmware Installed, Faster TELL1 Parameter Uploading, User/Expert Mode, Monitoring task for HLT triggers, Online Alignment Monitoring, Standalone Macro Package, Vetra Test Suite, Offline test of Closing with Fest'09 data</td>
</tr>
<tr>
<td>49</td>
<td>4/12</td>
<td>V7r0 python-ized Release, v7r0 python, motion System in private DB, Pulse Shape Scans</td>
</tr>
<tr>
<td>50</td>
<td>11/12</td>
<td>First TELL1 parameters uploaded</td>
</tr>
<tr>
<td>3</td>
<td>15/1</td>
<td>(system down for cooling) Offline alignment monitoring, Closing test with Fest'09 in HLT</td>
</tr>
<tr>
<td>4</td>
<td>22/1</td>
<td>(system down for cooling) integrate Marathons, Round Robin NZS, Review of monitoring algorithms and macros</td>
</tr>
<tr>
<td>5</td>
<td>20/1</td>
<td>(system down for cooling)</td>
</tr>
<tr>
<td>6</td>
<td>5/2</td>
<td>(Global Com.), Review of online monitoring default pages, Green/Red light from alignment monitoring, Closing tested with motion</td>
</tr>
<tr>
<td>7</td>
<td>12/2</td>
<td>HV Scan Data, Timing Parameters Known, Tuning settings to understand error banks</td>
</tr>
<tr>
<td>8</td>
<td>19/2</td>
<td>(TELL1 Replacement)</td>
</tr>
<tr>
<td>9</td>
<td>25/2</td>
<td>Tested Parameters, TELL1 Parameter uploaded, HLT Alley for Halo tracks, Green/Red light alignment, Monitoring package for error banks</td>
</tr>
<tr>
<td>10</td>
<td>5/3</td>
<td>Alerts Scheme Table for LHCb Shifter, Test Pulse Strip Scan, Timing Parameters Uploaded, Gain Calibration, Online Presenter Messages to locate problems, Velo DQ flags summary</td>
</tr>
</tbody>
</table>

...
## Milestones Achieved

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Milestone</th>
<th>Person responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>31/10</td>
<td>Parameter upload Plan</td>
<td>Kurt</td>
</tr>
<tr>
<td>44</td>
<td>31/10</td>
<td>PVSS restructuring</td>
<td>Karol</td>
</tr>
<tr>
<td>44</td>
<td>31/10</td>
<td>Test of naming panel and diagram</td>
<td>Mark</td>
</tr>
<tr>
<td>47</td>
<td>20/11</td>
<td>Alignment online monitoring</td>
<td>Marco</td>
</tr>
<tr>
<td>47</td>
<td>20/11</td>
<td>User/Expert Mode</td>
<td>Stefano</td>
</tr>
<tr>
<td>48</td>
<td>27/11</td>
<td>Integrate Marathons</td>
<td>Mark</td>
</tr>
<tr>
<td>40</td>
<td>04/12</td>
<td>Initial Parameters Uploaded, First TELL1 parameters uploaded</td>
<td>Kurt</td>
</tr>
</tbody>
</table>
Status report

No such situation!
The main software tool used to analyze NZS data
(installed on the online cluster at the pit)

1) Noise calculation/monitoring
2) Time alignment study
3) Beetle pulse shape measurement
4) Many more...

One of the most important tasks performed by Vetra platform is the TELL1 processing parameters calculation/tuning (see next slide)

- Velo needs $\sim10^6$ processing parameters
- The quality of these parameters is critical for the data quality (TED data)
- The pedestal values and clusterisation thresholds proved to have the most significant influence on the ZS data
Vetra – status (2/3)

VELOCOND → Gaudi Job (Vetra)

→ Tell1 Emulation

↓

Analysis

↓

Tuned Processing Parameters

↓

XML file

→ XML2PVSS

→ Monitoring/verification job

Noise Data

←

Tell1

←

.getElementsByName("name")

←

Data

←

Gaudi Job (Vetra)
Current release (ST+Velo) features:

1) The latest TELL1 firmware
2) Capability of analyzing the data taken in round-robin scheme
3) All the information needed for the data processing is stored in the ‘private’ databases (e.g., VELOCOND for the VELO part)
4) Bit-perfectness testing suite – to check/confirm that the output of the TELL1 board is the same as the one from the Emulation (comparison is made at the level of cluster – raw - banks)
5) Many changes and updates of the monitoring packages driven by the TED data taken during the August and September runs
6) Specialized package with standard macros and scripts for shifters to make common tasks easy and automated (see later)
7) Pythonized options – will become the default for the next release (see next slide)
Vetra – configurables and python options

**Configurables**
- Main configuration file, `Configuration.py`, created
- Several dedicated configuration classes for each Vetra processing phase (TELL1 processing, monitoring, output)
- Profited from occasion to improve structure of Vetra sequences
- Consistency checks introduced

**Python options**
- Most of Vetra options “translated” to Python
- Re-structuring / improvements on occasion

Release of pythonized Vetra v7r0 by end of the year
Software versioning

**Backing up repository**
- All VELO software in CVS
- Recipes now also backed up in PVSS database
- PVSS: plain-text and VELO-specific files in CVS

**Tagging and releases**
- Frequent commits
- Packaged tagged at most on a weekly basis
Recipes

- Physics (default)
- Physics NTP
- Physics TP
- Physics TP NZS (new)
- Delay scan
- Delay scan NTP (new)
- TTCRx (new)
- TTCRx NTP (new)
- Threshold scan (new)
- Strip scan (new)
Timing studies

- Timing scans implemented and being tested

- New firmware releases under test

- TTCrx scan: for time alignment of sensors
Timing studies

Optimization of analogue receiver (Arx) digitization time:
Monitoring

- A lot of progress recently

- Monitoring packages fully integrated in Vetra and Brunel:
  - VELO monitoring in standard Brunel jobs
  - Vetra has been updated with latest software

- Scripts and macros are being developed to analyse data
  (see later)

- Wiki pages with documentation and HowTo’s being written/updated
- Contents has been rather stable since ~August
- VELO has discussed his wish list at a recent “histograms and monitoring” meeting
- New features to be exploited
Monitoring – algorithms packages

**VeloRecMonitors**
- Package for “high-level” (= ZS) data
- Monitoring based on clusters and tracks (2 algorithms)
  - all in CVS
- Extra algorithms included; e.g. for beam position monitoring
- First alignment monitoring algorithm released

**VeloDataMonitor**
- Package for NZS data
- Same as always. Stable

**VeloClusterDataMonitor & VeloTrackDataMonitor**
- “Old” monitoring packages presently in “drain mode”
- Will be totally replaced by VeloRecMonitors
- But still used for now in the online monitoring
**VetraScripts**

- New package introduced recently
- To collect scripts, macros, Python modules for monitoring and analysis
- Many additions / improvements expected in next couple of months

For the moment:

- Torkjell: several noise monitoring macros
- Kazu, Sadia: macros for timing studies

Examples of ongoing work:

- Abdi: python script to display noise versus bias voltage
- Abdi, James: additions to cluster-based monitoring
- Barinjaka: IV scan analysis scripts
Example (MC) distributions from VeloRecMonitors

Active chip links versus sensor
Entries: 422111
Mean x: 54.56
Mean y: 46.5
RMS x: 37.34
RMS y: 27.62

Number of strips per cluster versus sensor
Entries: 422111
Mean x: 54.56
Mean y: 1.269
RMS x: 37.34
RMS y: 0.4967

Cluster ADC values versus sensor
Entries: 422111
Mean x: 54.56
Mean y: 74.02
RMS x: 37.34
RMS y: 35.81

Cluster ADC values versus sampling index
Entries: 463657
Mean x: 6.925
Mean y: 25.44
RMS x: 39.44
RMS y: 2.907
Noise monitoring macros – example of GUI (1/2)
Noise monitoring macros – example of GUI (2/2)
High Voltage

- PVSS recipes available to automate IV scans
- Set initial voltage, target voltage, step, single or set of sensors
- A data file produced per sensor containing channel number, voltage, current, sensor temperature
- Analysis scripts under development …
**Firmware**

- New release with round-robin NZS fixed
- Initial data taken on one TELL1
- In the meantime test with all VELO: OK
- Note: documentation has improved (release notes and versioning)

**Round-robin of NZS**

- First tests of data production: OK

**MCMS**

- Algorithm performance evaluation ⇒ firmware bug fixed
- Corrected version at the pit
- Tests with latest version of Vetra
TELL1 – parameter uploading

- Quality of data significantly affected by not having tuned TELL1 parameters uploaded

- First tests of data taking with uploaded TELL1 parameters: OK

- More tests being done
PVSS libraries for converting XML into the data structures digested by the framework functions that modify recipes

Underlying XML parser is the one recently provided as a patch by ETM (vendor of PVSS)

New panel implemented (help from Johan - thanks):
  allows 1-button action to amend the recipes from XML
  - tested successfully last Thursday by uploading different clustering thresholds, reordering settings and digitization delays to several TELL1’s on the VELO A-side
  - amending the recipes from XML for a VELO half takes ~5 min (expected speed improvement of factor 1.5-2)

Not everything covered, but including other parameters now straightforward
Conclusions

- VELO software plans carefully … planned!

- Items on “critical path” identified, given highest priority

- Good and steady progress observed