Belle II Tracking Validation Update

F2F Tracking Meeting 2015 - Vienna

Thomas Hauth, Oliver Frost | 22. April 2015

INSTITUTE OF EXPERIMENTAL NUCLEAR PHYSICS (IEKP)

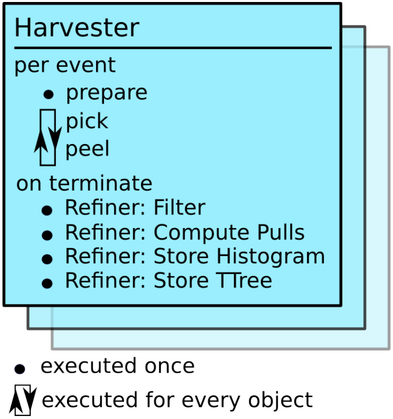
Validation TWiki page

TWiki page list some example command lines:

- Run tracking validation within basf2 validation framework
- Run standalone tracking validation
- Run standalone tracking validation with pre-generated events

https://belle2.cc.kek.jp/ twiki/bin/view/Software/TrackingValidation

Modular Validation Concept

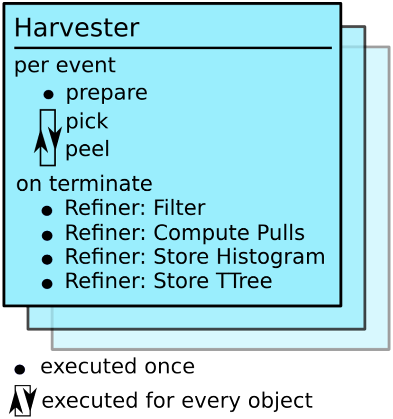


The naming scheme is adopted from the agriculture world ...

Harvester Component

- Can be configured to accept single Store-Obects (e.g. EventMetaData) or StoreArray entries (e.g. MCTrackCands)
- The `pick` function can accept or reject entries
- Relevant information is extracted in the `peel` function and stored in a dictionary, called *crops*
- Multiple harvester can run: MCTrackCands and EventMetaData for example

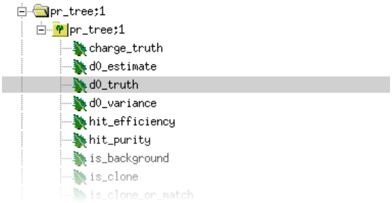
Modular Validation Concept



Refiner Component

- Refiner can either post-process, sort or save crops
- Refiner are not depending a specific harvester
- One or more refiner can be registered to run as part of one Harvester

TTree output option



```
pr_tree;1
  pr_tree;1
    charge_truth
    d0_estimate
    d0_truth
    d0_variance
    hit_efficiency
    hit_purity
    is_background
    is_clone
    is_clone_or_match
```

- Extracted validation quantities can be easily stored in a root tree
- Three trees are stored at the moment:
  - MC tracks (with matching information)
  - PR tracks (with matching information: MC track, is fake, is background etc.)
  - Event information (number of MC tracks, number of MC particles etc.)
- Note: the validation histograms and graphs are not created from the TTree content, but directly from the crops
- Nota bene: for more complex studies (following MC ¡-¿ PR relations, studying hit relations etc.) we recommend to implement custom validation module

TTree output option

Two examples of quick 'n dirty debugging using the quad tree and the ROOT interactive console: Compute the ϕ residual only for positively charged tracks:

```
pr_tree->Scan("phi0_estimate-phi0_truth","charge_truth == 1");
```

Plot a histogram of the number of PXD hits of all tracks which have parameter $\|z_0\| > 0.5$:

```
pr_tree->Draw("n_pxd_hits", "TMath::Abs(z0_truth)>0.5");
```

Modular Validation Concept

Advantages

- The components to extract (Harvester) and process/store validation quantities is independant and modular
- Complex tasks like resolution determination or creating TTrees only need to be implemented once and applied for many quantities
- Increased maintainability and less amount of code in the validation codebase

Disadvantages

- Programmers of validation scripts need to understand the concept behind harvester/refiner
  - Remedy: some documentation and examples are provided, can be improved
- Uses-cases may arise which are not covered by the current design
  - Remedy: extend functionality if necessary

Modular Validation Concept - Current status

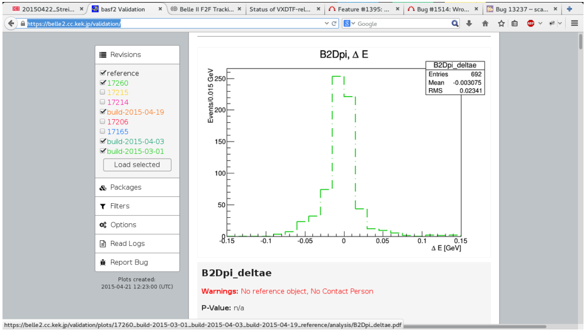
- Oli's implementation is available in the basf2 subversion
- It implements all the tracking validation plots of the 'old' validation
- Execute with:

```
python tracking/scripts/tracking/validation/module.py
```

and have a look at the file `test_separated_module.root`

Status new validation website

https://belle2.cc.kek.jp/validation/



Plots created:
2015-04-21 12:23:00 (UTC)

B2Dpi_deltae

Warnings: No reference object, No Contact Person

P-Value: n/a

https://belle2.cc.kek.jp/validation/plots/17260_build-2015-03-01_build-2015-04-03_build-2015-04-19_reference/analysis/B2Dpi_deltae.pdf

- Progress on the new validation website was not as fast as hoped ...
- Open issues/requests are listed on the redmine website
- Timothy has started to work on open issues yesterday

Discussion