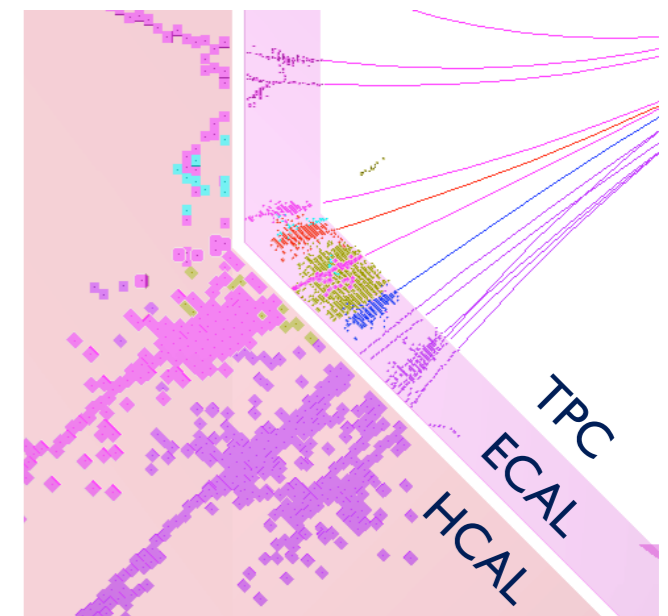




Pandora Update

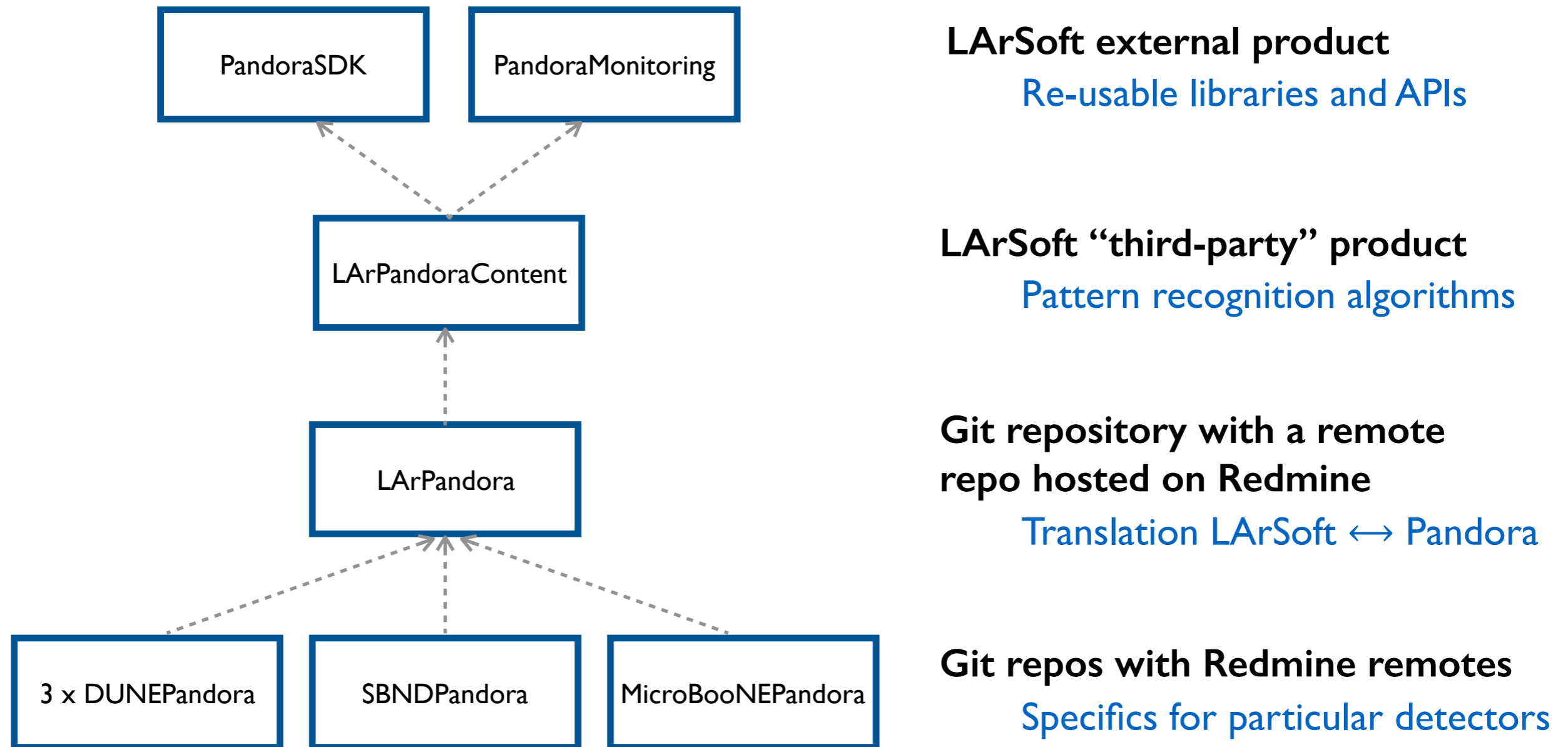
J. S. Marshall
LArSoft Coordination Meeting
8 March 2016





Pandora LArSoft Integration

Simple cartoon showing the Pandora packages and an indicative hierarchy:

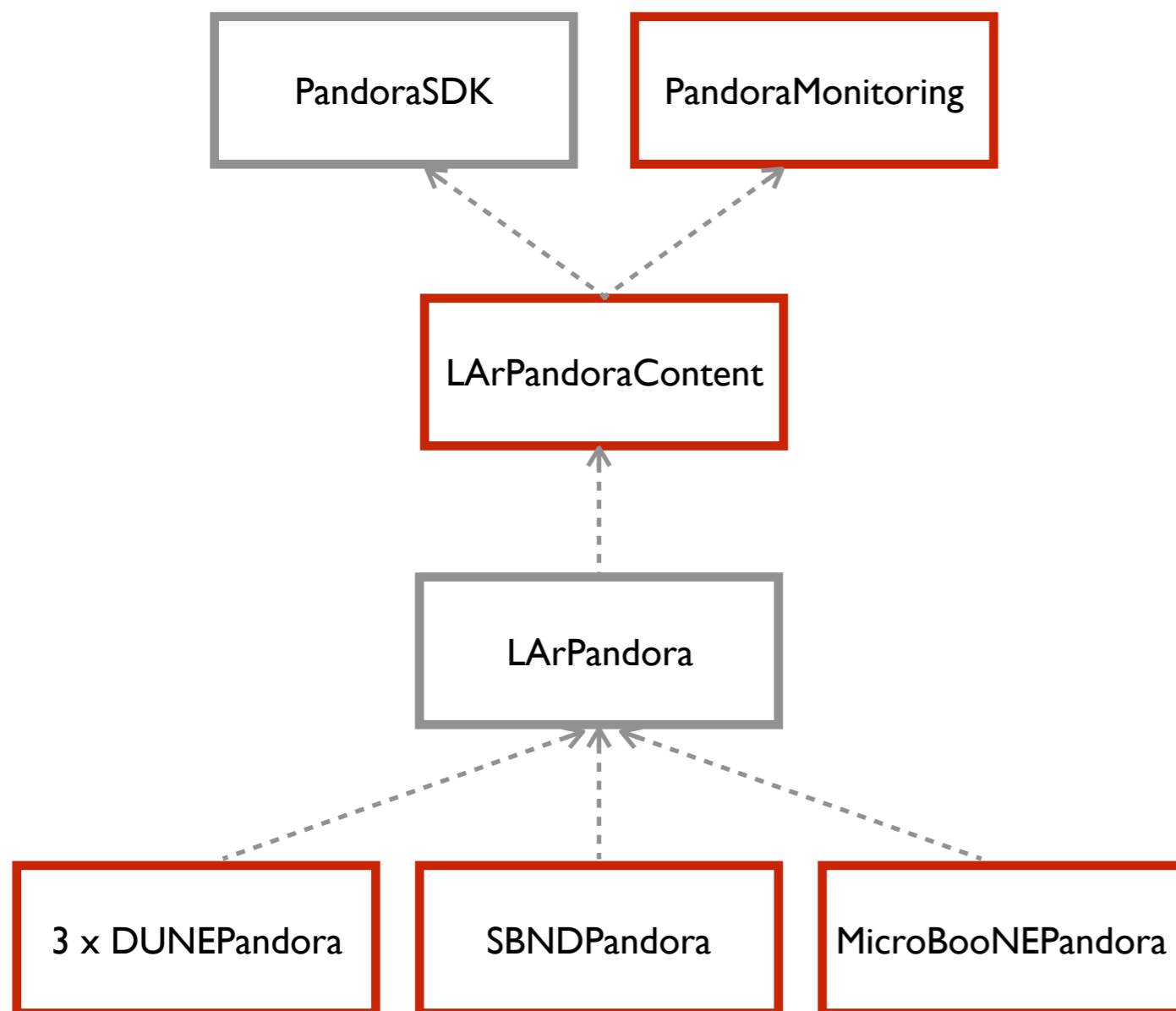


Pandora’s home on the web: <https://github.com/PandoraPFA>



Pandora LArSoft Integration

List of action items, as presented in the last coordination meeting



LArSoft external product

ROOT-based PandoraMonitoring not fully functional within LArSoft

[LArSoft “third-party” product]

Make more accessible to LArSoft users e.g. immediate access to feature branches

Git repository with a remote repo hosted on Redmine

[**Git repos with Redmine remotes**]

Probably unnecessary, if suitable geometry abstraction is available

See <https://indico.fnal.gov/conferenceDisplay.py?confId=11554>



TApplication Issues

ROOT-based PandoraMonitoring not fully functional within LArSoft

<https://cdcv.sfnal.gov/redmine/issues/11659>

- PandoraMonitoring uses the ROOT Event Visualisation Environment (TEVE), which requires access to a “full” TApplication, with an interface to the graphics system, etc.
- Unfortunately this setup is affected by a race for TApplication resources:
 - Rather easy for a user to implicitly create just a “default/basic” TApplication (which becomes the “gApplication”), e.g. via TROOT::Idle(...), TROOT::ProcessLine(...), etc.
 - If a “default” TApplication already exists, graphics support is unavailable. ROOT certainly used to have fatal problems reconfiguring/replacing existing TApplication to pick-up graphics, etc.
 - Need to get there first to create a full TApplication with graphics support.

Default behaviour:

```
<snip>
TimeModule> run: 1 subRun: 0 event: 1 cctrack CCTrackMaker 0.00292397
PandoraMonitoring, only able to use default TApplication (limited functionality).
PandoraMonitoring::InitializeEve(): DISPLAY environment set to :1044.0
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create ROOT is running in batch mode.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create window system not initialized.
Failure in algorithm 0x4f6d690, LArVisualMonitoring, unrecognized exception
</snip>
```



TApplication Issues

- Gianluca has kindly pushed a feature branch to the larpandora Redmine remote: `feature/gp_PandoraVisualization`, which includes a new `RootGraphicsEnablingService`.
- Elegant way to be the first to the TApplication resources, explicitly creating a full TApplication with all required resources, before other operations could create only a “default” TApplication.
- In-Pandora visualisation tools now work nicely in LArSoft. Enables full support for Pandora algorithm development in LArSoft.

Interface:

```
/**
 * This class implements a work around to initialize ROOT graphics system
 * before some other code messes with it (that is, we are the first ones
 * to mess with it, and of course we do it right(TM)).
 * To ensure this code is executed as soon as possible, it's bound to
 * the construction of a static variable. While it is not predictable
 * when this code will be executed, it is expected to be executed before
 * main() function of the executable if linked statically, or as soon
 * as a library statically linked with this one is dynamically loaded.
 *
 * The work around consists of making sure there is an active TApplication
 * (probably any will do), pulling ROOT out of batch mode, and creating
 * ROOT's X11 graphics client (that, for example, TEve needs).
 */
struct RootGraphicsEnablerClass
{
    /// Default constructor: quiet
    RootGraphicsEnablerClass() { EnableRootGraphics(); }

    /// Constructor: enable messages on specified stream
    RootGraphicsEnablerClass(std::ostream& out) { EnableRootGraphics(&out); }

    /// Enacts the tricks to enable the graphics
    /// @param out pointer to output stream (default: none, be quiet)
    static void EnableRootGraphics(std::ostream* out = nullptr);
};
```

In .fcl file:

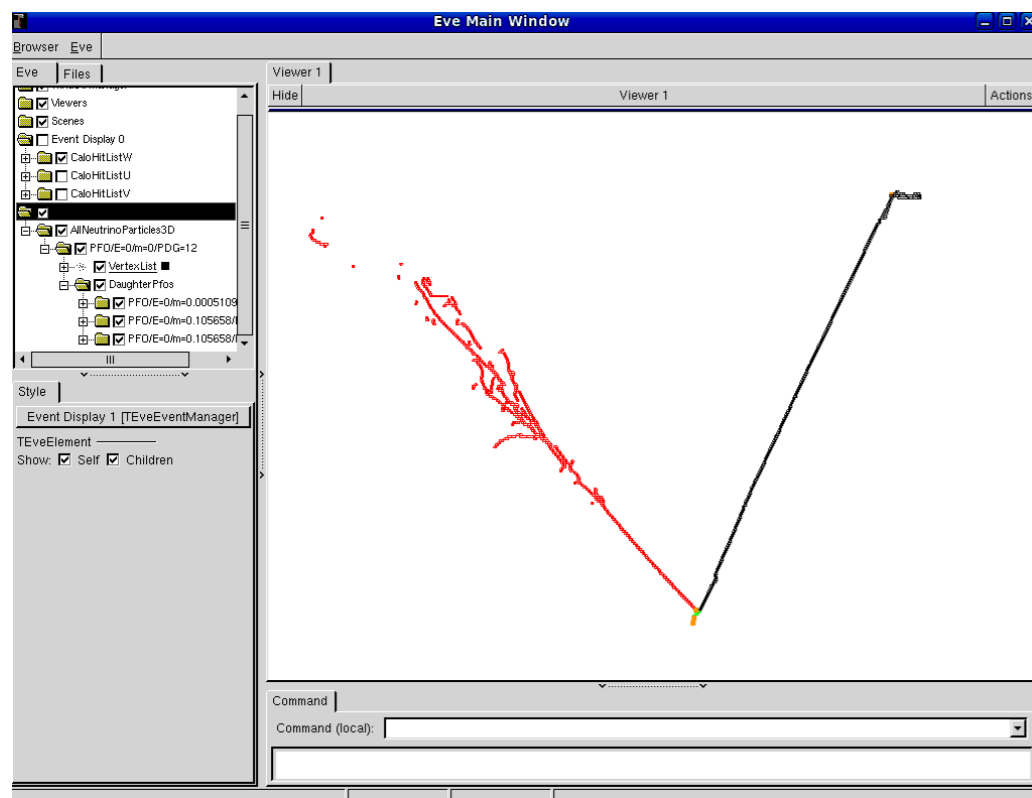
```
services:
{
  RootGraphicsEnablingService: {}
  ...
}
```

In PandoraSettings.xml file:

```
<pandora>
  <IsMonitoringEnabled>true</IsMonitoringEnabled>
  ...
  <algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListW</CaloHitListNames>
  </algorithm>
  ...
</pandora>
```




Pandora In-Algorithm Visualisation



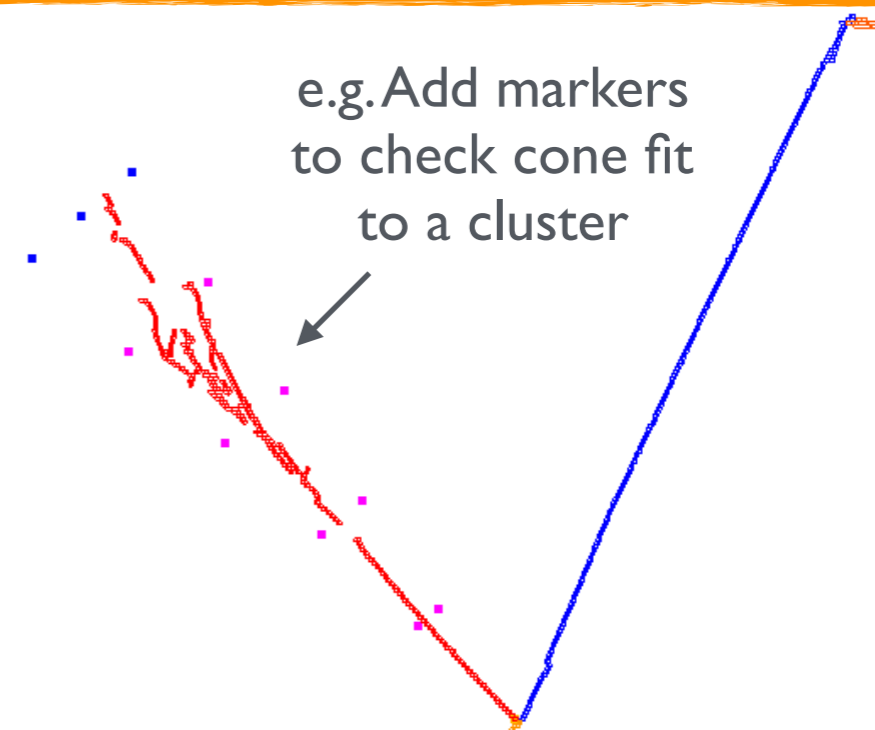
```

. . .
<algorithm type = "LARLayerSplitting"/>
<algorithm type = "LARLongitudinalAssociation"/>
<algorithm type = "LARVisualMonitoring">
  <ClusterListNames>ClustersU</ClusterListNames>
</algorithm>
<algorithm type = "LARTransverseAssociation"/>
<algorithm type = "LARVisualMonitoring">
  <ClusterListNames>ClustersU</ClusterListNames>
</algorithm>
<algorithm type = "LARLongitudinalExtension"/>
<algorithm type = "LARTransverseExtension"/>
<algorithm type = "LAROvershootSplitting"/>
<algorithm type = "LARBranchSplitting"/>
<algorithm type = "LARKinkSplitting"/>
. . .

```

e.g. Add two event display algs to examine changes as reconstruction progresses

- ROOT Event Visualisation Environment provides 2D or 3D displays of hits, clusters, particles or vertices.
- Provides support for visualisation of navigable particle hierarchies, custom markers, colour schemes, etc.
- Reusable Pandora event display alg can be added to XML to view status of reconstruction at any point.
- Alternatively, algs can use visualisation APIs to provide custom visual debugging - rewarding way to work.





Pandora Validation Tools



Comprehensive list of reco particle to MC primary matches

Clickable list of objects for each MC primary
Indication of reco quality: "angry" colours for poor matches

```

> Running Algorithm: 0x2f26850, LArEventValidation
---RAW-MATCHING-OUTPUT---
MCNeutrino, PDG 14, Nuance 1004
RecoNeutrino, PDG 12
VtxOffset x: -0.00564575 y: -0.0757809 z: 0.383545 length: 0.391

Primary 0, PDG 22, nMCHits 724 (183, 298, 243)
-MatchedPfo 0, PDG 11, nMatchedHits 724 (183, 298, 243), nPfoHits 1196 (325, 486, 385)

Primary 1, PDG 13, nMCHits 684 (134, 348, 202)
-MatchedPfo 1, PDG 13, nMatchedHits 652 (123, 332, 197), nPfoHits 667 (128, 338, 201)

Primary 2, PDG 22, nMCHits 438 (141, 155, 142)
-MatchedPfo 0, PDG 11, nMatchedHits 438 (141, 155, 142), nPfoHits 1196 (325, 486, 385)

Primary 3, PDG 2112, nMCHits 67 (21, 24, 22)

Primary 4, PDG 2212, nMCHits 58 (16, 30, 12)
-MatchedPfo 2, PDG 13, nMatchedHits 54 (14, 28, 12), nPfoHits 54 (14, 28, 12)
-MatchedPfo 0, PDG 11, nMatchedHits 1 (1, 0, 0), nPfoHits 1196 (325, 486, 385)
-MatchedPfo 1, PDG 13, nMatchedHits 1 (1, 0, 0), nPfoHits 667 (128, 338, 201)

Primary 5, PDG 2212, nMCHits 15 (4, 7, 4)
-MatchedPfo 1, PDG 13, nMatchedHits 14 (4, 6, 4), nPfoHits 667 (128, 338, 201)

Primary 6, PDG 2112, nMCHits 0 (0, 0, 0)

Primary 7, PDG 2212, nMCHits 0 (0, 0, 0)
---PROCESSED-MATCHING-OUTPUT---
Primary 0, PDG 22, nMCHits 724 (183, 298, 243)
-MatchedPfo 0, PDG 11, nMatchedHits 724 (183, 298, 243), nPfoHits 1196 (325, 486, 385)

Primary 1, PDG 13, nMCHits 684 (134, 348, 202)
-MatchedPfo 1, PDG 13, nMatchedHits 652 (123, 332, 197), nPfoHits 667 (128, 338, 201)

Primary 2, PDG 22, nMCHits 438 (141, 155, 142)

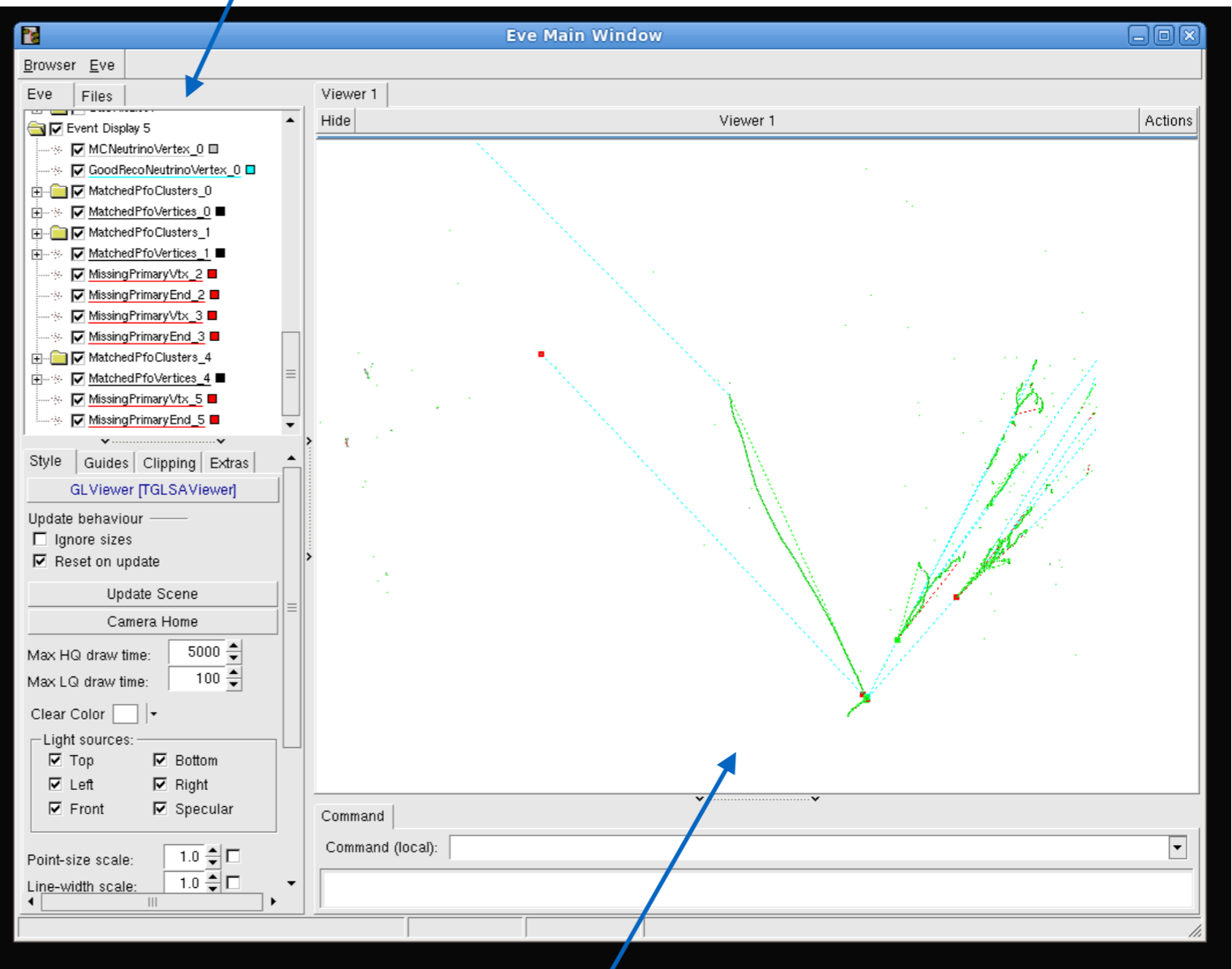
Primary 3, PDG 2112, nMCHits 67 (21, 24, 22)

Primary 4, PDG 2212, nMCHits 58 (16, 30, 12)
-MatchedPfo 2, PDG 13, nMatchedHits 54 (14, 28, 12), nPfoHits 54 (14, 28, 12)

Primary 5, PDG 2212, nMCHits 15 (4, 7, 4)

Is correct? 0
Press return to continue ...

```



Output of our interpretative matching scheme

Matched particles appear in green
Split particles appear in red
Red markers placed at vtx/end of missing particles

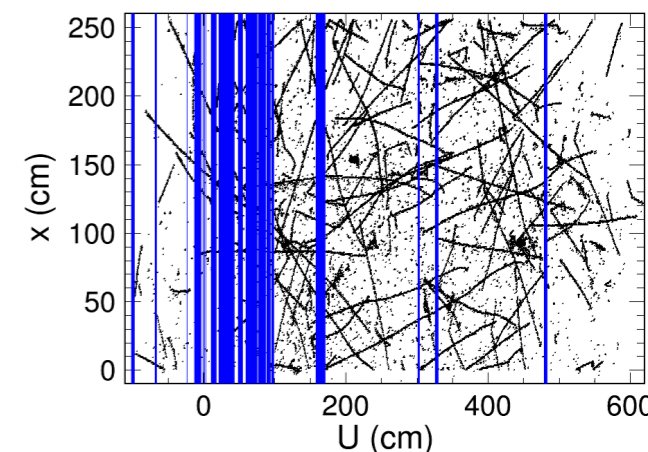


Pandora Recent/Upcoming Changes

- Intend to make a new Pandora release soon, probably Wed/Thu this week:
 - Will include some very minor changes to PandoraSDK and PandoraMonitoring, aligning the versions with those in current use elsewhere e.g. ILCSoft.
 - These changes will not impact the LArSoft usage and are not breaking changes. (Some very minor changes under the hood and some minor new functionality).
- Includes/is-accompanied-by new version of LArPandoraContent algorithm library. Will use some new PandoraMonitoring functionality and include some new algs and updates:
 - Parameter optimisation for `VertexSelection` algorithm by Joris (Cambridge grad. student).
 - New `SplitShowerMerging` algorithm, derived from existing `VertexBasedPfoMerging` algorithm. Helps to merge-together any split showers remaining in Pandora neutrino pass.
 - Improved tools for hand-scanning events, with highlighting of pattern recognition problems.

- Advance notice of upcoming change (no firm timescale):
 - Introduce new `LineGap` class into the PandoraSDK (already supports `BoxGaps` and `ConcentricGaps` for ILC/LHC). Interface:

```
bool IsInGap(const CartesianVector &, const float gapTolerance) const;
```
 - Use in one or two important 2D reco, cluster-merging algorithms.

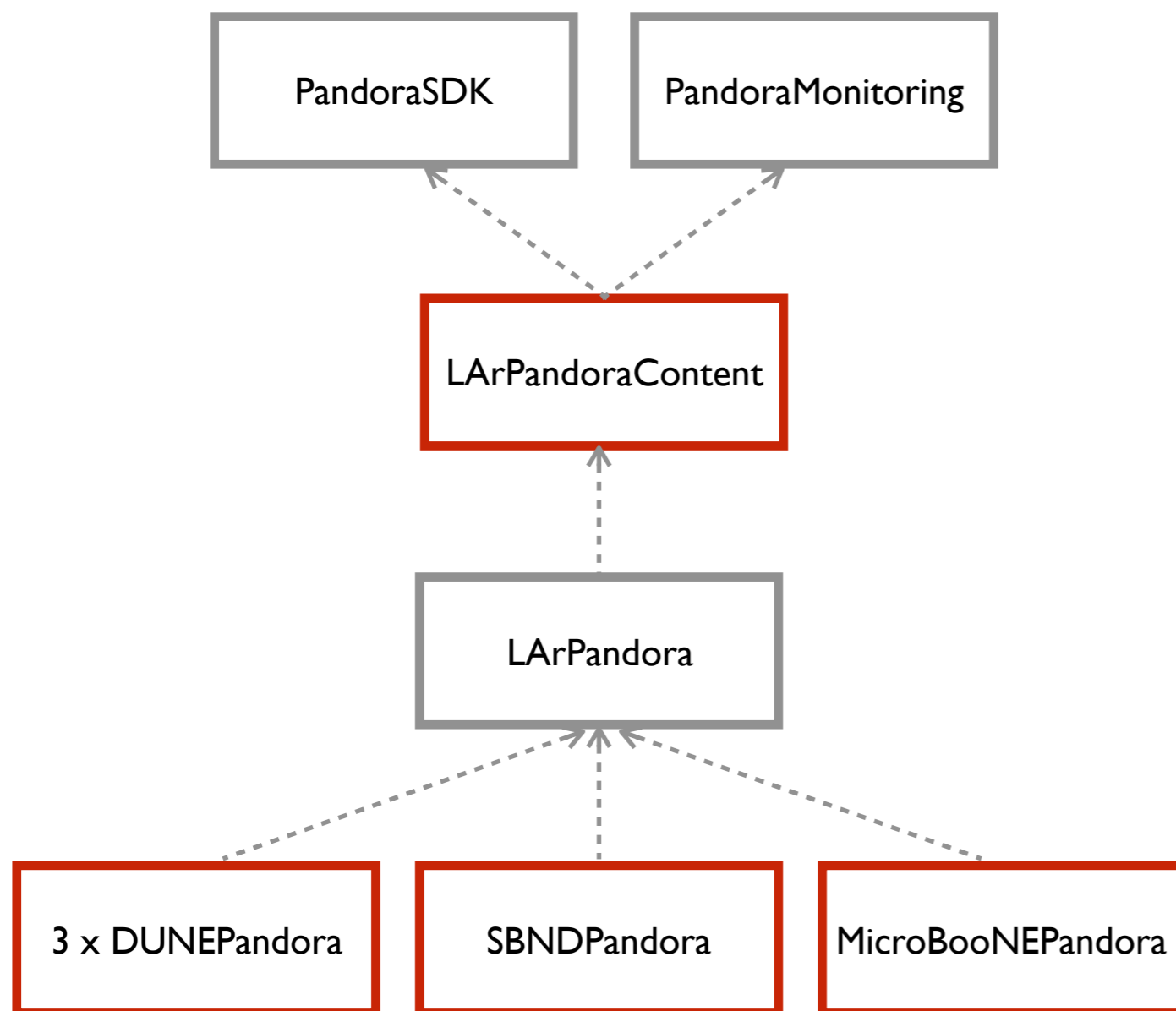




Pandora LArSoft Integration



Remaining action items



LArSoft external product

Continued discussion

[LArSoft “third-party” product]

Make more accessible to LArSoft users
e.g. immediate access to feature branches

Git repository with a remote
repo hosted on Redmine

Probably simple: not yet investigated

[Git repos with Redmine remotes]

Probably unnecessary, if suitable
geometry abstraction is available

Plus, understand status of any investigations into interplay between Pandora and Track3DKalmanHit



Thanks for your attention!



Pandora Contact Details



**Pandora is an open project and new contributors would be extremely welcome.
We'd love to hear from you and we will always try to answer your questions!**

Contact details:

Framework development

John Marshall (marshall@hep.phy.cam.ac.uk)
Mark Thomson (thomson@hep.phy.cam.ac.uk)

LAr TPC algorithm development

John Marshall
Andy Blake (a.blake@lancaster.ac.uk)

Performance metrics and validation

John Marshall
Andy Blake
Lorena Escudero (escudero@hep.phy.cam.ac.uk)
Joris Jan de Vries (jjd49@hep.phy.cam.ac.uk)
Jack Weston (weston@hep.phy.cam.ac.uk)

Please visit <https://github.com/PandoraPFA>

