Proposed Cooker Analysis Flow

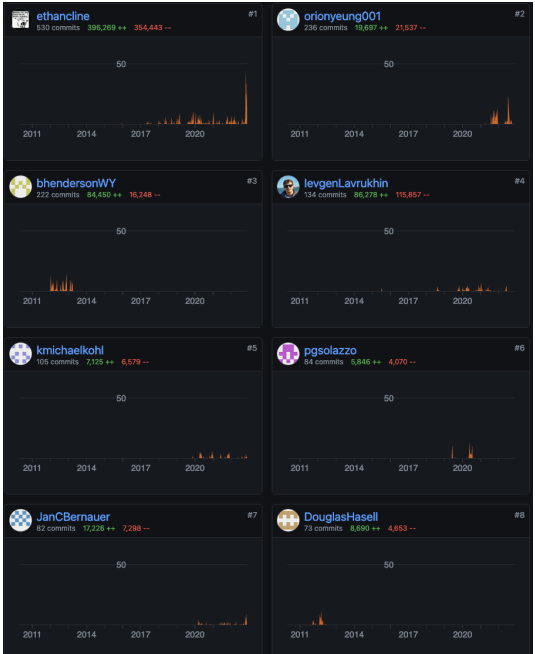
Ethan Cline & Bishoy Dongwi

December 3, 2022

The Cooker

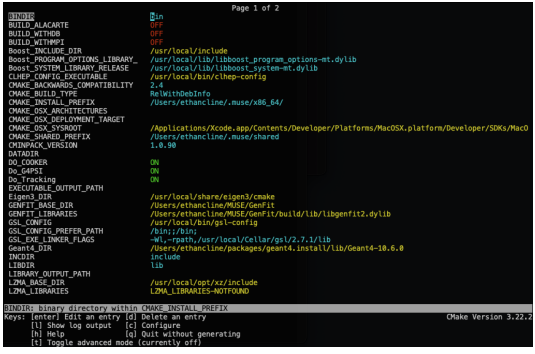
- History
  - Originally written for OLYMPUS by Jan
  - Ported to MUSE by Jan
  - Also brought to TREK by Bishoy
- Hosted on GitHub, continuous history

Other Members of DarkLight have some experience



Some Technical Details

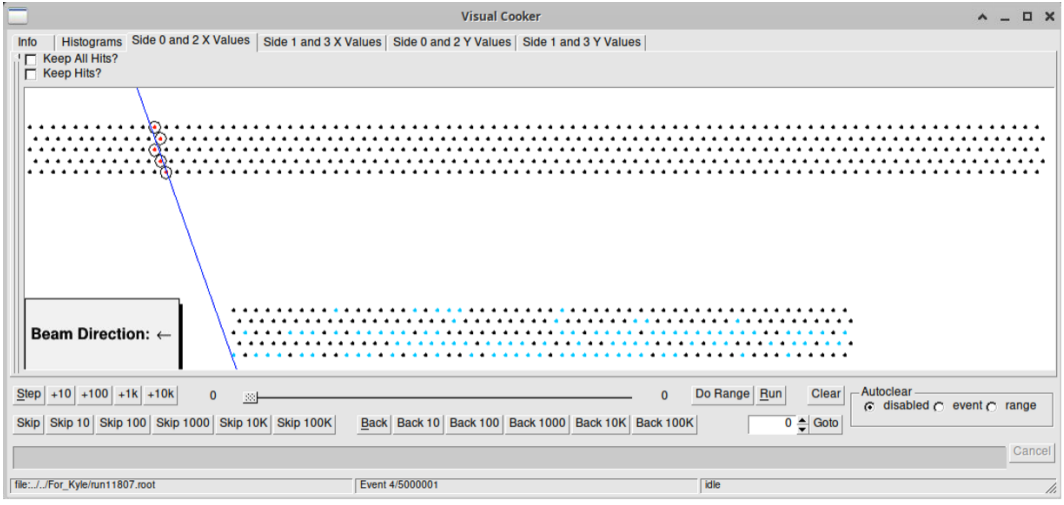
- Keyword: Modular
- Very thin layer on top of ROOT
- cmake based compilation (ccmake is nice)
  - Control over your compilation options
- Cordon off development of expert features
- Leave everything easily accessible when necessary
- Simulation can be included!



Useful Features

- Event Display built-in
- Visual cooker can handle event by event analysis in GUI
- Can write run info to remote database
- Base “Plugin” class provides useful rapid prototyping histogram functionality
- Easily extensible to handle external libraries (GenFit for MUSE)
- Can pass arbitrary commands from command line to any plugin
- Framework for handling individual config files
- Already have a MIDAS reader available

Visual Cooker



A Cooker Command

cooker gemini.xml run13594.root run13594_gemini.root

```
<?xml version="1.0" encoding="UTF-8" ?>
<muse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation='recipe.xsd'>

  <init>init/GEMini.xml</init>

  <source>MMT</source> <!-- the source tree name-->
  <destination>lumigencooked</destination> <!-- the destination tree name-->

  <plugins>
    <plugin>
      <name>GEMini</name>
      <file>libGEMini</file>
    </plugin>
  </plugins>

  <!-- These functions will be called at the 'very' beginning -->
  <defineHistograms>
  </defineHistograms>

  <!-- These functions will be called at the beginning -->
  <startup>
    <GEMini>plotting</GEMini>
    <GEMini>startup</GEMini>
  </startup>

  <!-- These functions will be executed for every event, in order of appearance -->
  <execute>
    <GEMini>process</GEMini>
  </execute>

  <postprocess>
  </postprocess>

  <!--second pass through the data-->
  <execute2>
  </execute2>

  <!-- These functions will be called at the end of the file -->
  <finalize>
    <GEMini>findPedestals</GEMini>
    <GEMini>findGains</GEMini>
  </finalize>

</muse>
```

A recipe

A Cooker Recipe


```
<?xml version="1.0" encoding="UTF-8" ?>
<muse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation='recipe.xsd'>

  <init>init/GEMini.xml</init>

  <source>MMT</source> <!-- the source tree name-->
  <destination>lumigemcooked</destination> <!-- the destination tree name-->

  <plugins>
    <plugin>
      <name>GEMini</name>
      <file>libGEMini</file>
    </plugin>
  </plugins>

  <!-- These functions will be called at the 'very' beginning -->
  <defineHistograms>
  </defineHistograms>

  <!-- These functions will be called at the beginning -->
  <startup>
    <GEMini>plotting</GEMini>
    <GEMini>startup</GEMini>
  </startup>

  <!-- These functions will be executed for every event, in order of appearance -->
  <execute>
    <GEMini>process</GEMini>
  </execute>

  <postprocess>
  </postprocess>

  <!--second pass through the data-->
  <execute2>
  </execute2>

  <!-- These functions will be called at the end of the file -->
  <finalize>
    <GEMini>findPedestals</GEMini>
    <GEMini>findGains</GEMini>
  </finalize>
```

Multi-Plugin Recipe

```
<?xml version="1.0" encoding="UTF-8" ?>
<muse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation='recipe.xsd'>

<init>init/tracking.xml</init>

<source>lumigemcooked</source> <!-- the source tree name-->
<destination>GEMTracks</destination> <!-- the destination tree name-->

<plugins>
   <plugin>
      <name>GeomLoader</name>
      <file>libGeomLoader</file>
   </plugin>
   <plugin>
      <name>GEM_GFT</name>
      <file>libGEM_GFT</file>
   </plugin>
</plugins>


<!-- These functions will be called at the 'very' beginning -->
<defineHistograms>
</defineHistograms>

<!-- These functions will be called at the beginning -->
<startup>
   <GeomLoader>startup</GeomLoader>
   <GEM_GFT>startup</GEM_GFT>
</startup>

<!-- These functions will be executed for every event, in order of appearance -->
<execute>
 <GEM_GFT>process</GEM_GFT>
</execute>

<!-- These functions will be called after the first pass of data -->
<postprocess>
</postprocess>

<!-- These functions call for a 2nd pass of the data and will be executed for every event -->
<execute2>
</execute2>

<!-- These functions will be called at the end of the file -->
<finalize>
```

Multiple Plugins in one recipe

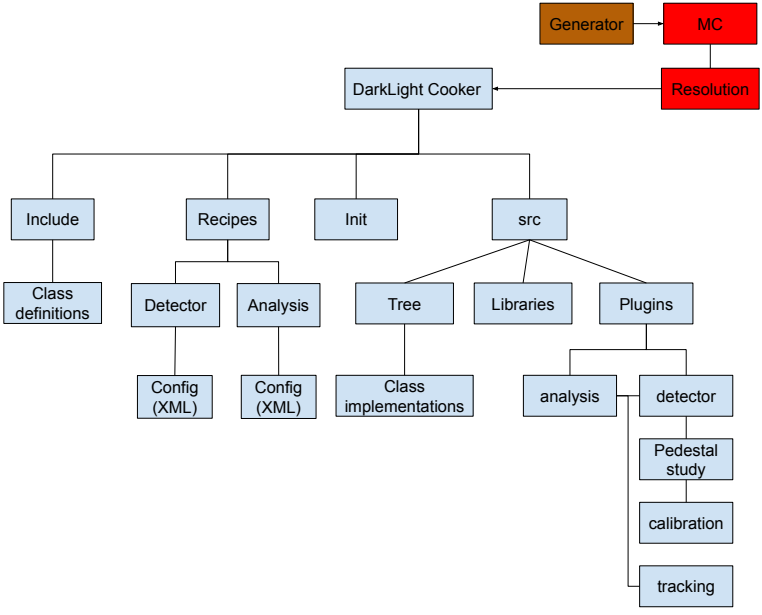
Multiple Functions from multiple plugins

Multi-Input Recipe

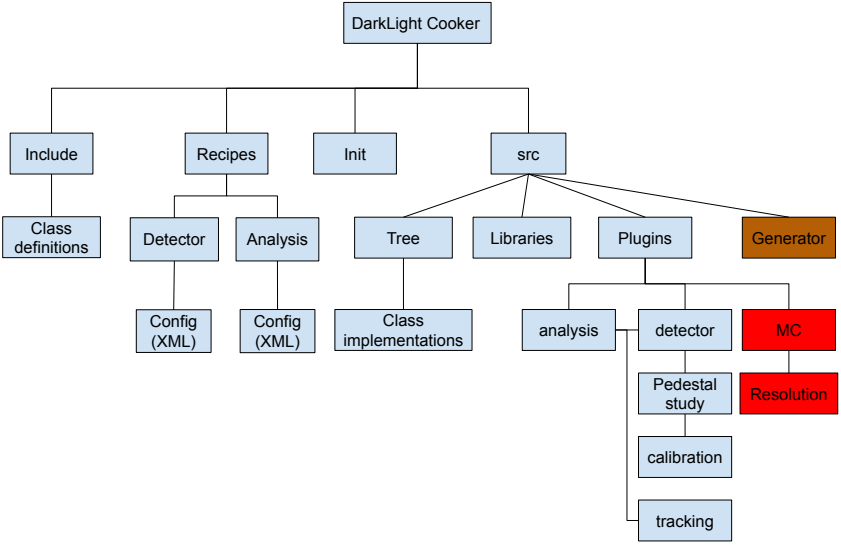
```
<?xml version="1.0" encoding="UTF-8" ?>
<muse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation='recipe.xsd'>

<!-- <init>init/all.xml</init> -->
<init>init/tracking.xml</init>

<source>BH:lumigemcooked</source> <!-- the source tree name-->
<!-- <source>BH:MUSEY:lumi:BEM</source> --> <!-- the source tree name-->
<destination>GEMcorrelation</destination> <!-- the destination tree name-->

<plugins>
  <plugin>
    <name>GEM_corr</name>
    <file>libGEM_corr</file>
  </plugin>
</plugins>


<!-- These functions will be called at the 'very' beginning -->
<defineHistograms>
</defineHistograms>

<!-- These functions will be called at the beginning -->
<startup>
 <GEM_corr>startup</GEM_corr>
</startup>

<!-- These functions will be executed for every event, in order of appearance -->
<execute>
 <GEM_corr>process</GEM_corr>
</execute>

<!-- These functions will be called after the first pass of data -->
<postprocess>
</postprocess>

<!-- These functions call for a 2nd pass of the data and will be executed for every event -->
<execute2>
</execute2>

<!-- These functions will be called at the end of the file -->
<finalize>
  <GEM_corr>finalize</GEM_corr>
</finalize>
```

Multiple input files

Current Layout - Multiple Repos



Proposed Layout - Single Repo



Simplify Flow

Everyone* who does analysis will need all parts of the analysis, cooker, Geant4, event generators – why keep them separate? Increases overhead.

Anything you don't want to use, don't compile. Cmake switch can disable unwanted parts.

Can still have groups work on different projects, separated into plugins, controlled via git.

*Those doing Ph.D. or master's projects, or otherwise writing a paper on the physics results

Dependencies

- Cooker Dependencies (Can be reduced)
  - ROOT
  - libboost
  - Xqilla
  - CLHEP
  - libXML
  - (FTGL)
  - (GSL)
  - LZMA
- ROOT Dependencies
  - Cannot be installed via package manager
  - XML
  - GDML
- Geant4 Dependencies
  - Visualization (QT? X11?)
  - GDML
- Cedar modules
  - StdEnv/2020 gcc/9.3.0
  - geant4/10.7.3
  - clhep/2.4.4.0
  - root/6.24.06
  - xerces-c++/3.2.2
  - gsl
  - boost/1.72.0

Case Study on Unification - Geometry

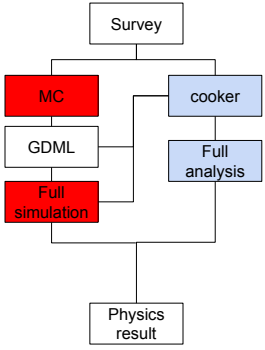
How do we incorporate survey information efficiently into the analysis and in sync with the simulation?

GDML: Geometry Design Markup Language. Designed as ROOT - Geant4 interface.

Monte Carlo is the "Source of Truth" for position of all detectors.

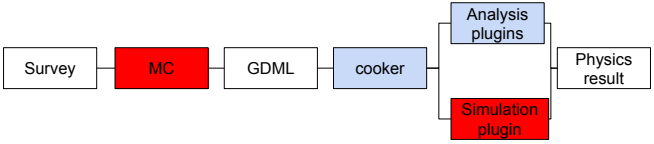
How do we keep things synchronized and smoothly flowing? Make simulation a plugin!

Case Study on Unification - Geometry



Two parallel analysis paths feeding from MC into cooker and both into the final result

Case Study on Unification - Geometry



MC is source of geometry truth which it provides to the cooker through the library and plugin functionality

With cooker config XML files, and a geometry library that loads the simulation detector construction, we create our GDML files on the fly every time we run cooker. This allows us to specify the geometry on a run-by-run basis!

Case Study on Unification - Geometry

What does the end user need to do to load the geometry?

Add the right header, load the geometry object with a one-liner in the plugin and modify the recipe

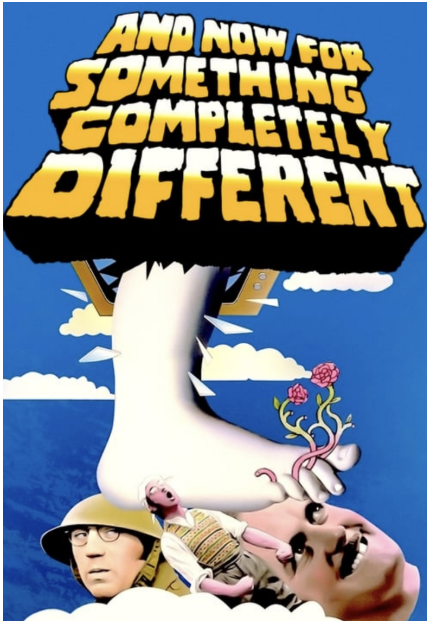
```
<?xml version="1.0" encoding="UTF-8" ?>
<muse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation='recipe.xsd'>

<init>init/tracking.xml</init>

<source>lumigemcooked</source> <!-- the source tree name-->
<destination>GEMTracks</destination> <!-- the destination tree name-->

<plugins>
  <plugin>
    <name>GeomLoader</name>
    <file>libGeomLoader</file>
  </plugin>
  <plugin>
    <name>GEM_GFT</name>
    <file>libGEM_GFT</file>
  </plugin>
</plugins>


<!-- These functions will be called at the 'very' beginning -->
<defineHistograms>
</defineHistograms>

<!-- These functions will be called at the beginning -->
<startup>
  <GeomLoader>startup</GeomLoader>
  <GEM_GFT>startup</GEM_GFT>
</startup>

<!-- These functions will be executed for every event, in order of appearance -->
<execute>
 <GEM_GFT>process</GEM_GFT>
</execute>

<!-- These functions will be called after the first pass of data -->
```

Multiple Plugins in one recipe

Multiple Functions from multiple plugins

Final Thoughts

- Have not begun merging simulation and cooker without collaboration OK
- Should include event generators as well
  - Don't add significant dependencies (compared to ROOT/Geant4)
  - Can specify which generator, what configuration via command line arguments, or recipes
- cmake keeps things separated and modular so if you have a student who wants to only touch one part, they don't need to have everything installed


AND NOW FOR
SOMETHING
COMPLETELY
DIFFERENT

Call for workshop proposals for DNP Hawaii 2023 Meeting

Dear Colleagues,

The DNP is pleased to announce that the 6th Joint Meeting of the Nuclear Physics Divisions of the APS and JPS will take place October 8-12, 2023 in Hawaii.

One integral part of the Hawaii meetings has been joint pre-meeting workshops. This announcement solicits proposals for these workshops. Please consider organizing a workshop or disseminating this solicitation to others who may be interested in organizing a workshop. For reference, the 2018 workshops are listed on the APS website.

We are expecting to be able to hold up to 30 in-person workshops. Each workshop is required to have one APS and one JPS organizer. Each workshop includes 6 speakers (3 from JPS and 3 from APS) that the workshop organizers should identify. Examples of workshop proposals can be found here and as well as here. Please fill out the Google form with the information. (Note that the form does not have to be filled out all at once, it is possible to go back and make edits.)

If you have any questions, please contact Baha Balantekin, Kouichi Hagino, or myself.

Thank you for taking the time to help make this part of the Hawaii meeting a great success. The forms should be filled out with proposed speakers and talk titles by **January 15, 2023**. A final selection will be carried out at this time and acceptance notifications sent out in early February.

Best Regards,

Ramona Vogt
DNP Secretary/Treasurer