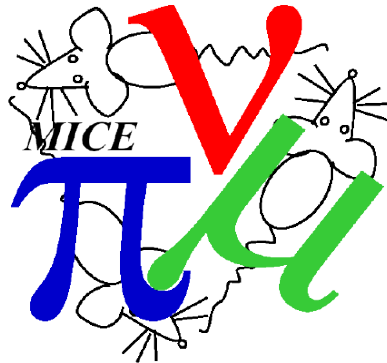




MAUS Overview

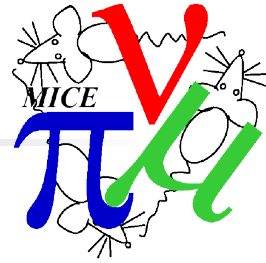


Chris Rogers,
ASTeC,
Rutherford Appleton Laboratory



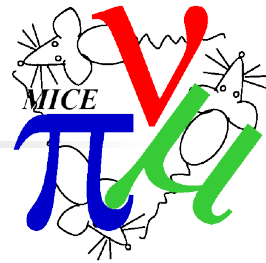


Overview



- Comment on ownership of code blocks
- Documentation update
- Test coverage update
- Code infrastructure status
- Detector reconstruction status

Responsibility/Block Diagram



Project management
Rogers

Same as in CM32!

Build system
Rogers

QA
Rogers

Documentation
Rogers

Geometry + fields
Rogers/Littlefield

Geant4 Simulation
Rogers

Data flow/API
Rogers/Richards

TOF
Rajaram

Tracker
Dobbs/Santos et al

Ckov
Cremaldi/Kafka

KL
Bogomilov

Data Unpacking
Karadzhov

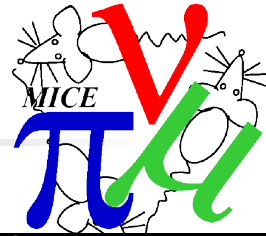
EMR
Karadzhov/Ruslan

RF
Smith

Detector Integration
Rogers/Lane

Accelerator physics
analysis
Rogers/Lane

Documentation



Applications Places Tue 26 Jun, 6:23 AM

maus_user_guide - Mozilla Firefox

maus_user_guide

micewww.pp.rl.ac.uk/maus/MAUS_latest_version/maus_user_guide/index.html

Most Visited Linux Mint Community Forums Blog News

MAUS Analysis User System

User Guide

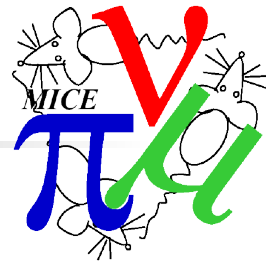
- [Contents](#)
- [What Who and How?](#)
 - [Who Should Use MAUS](#)
 - [Getting the Code and Installing MAUS](#)
 - [Running MAUS](#)
 - [Run Control](#)
 - [Other Applications](#)
 - [Accessing Data](#)
 - [Loading ROOT Files in Python Using PyROOT](#)
 - [Loading ROOT Files on the ROOT Command Line](#)
- [Using and Modifying the Data Structure](#)
 - [Accessing ROOT files](#)
 - [Conversion to, and Working With, JSON](#)
 - [Extending the Data Structure](#)
- [Running the Monte Carlo](#)
 - [Beam Generation](#)
 - [GEANT4 Bindings](#)
- [Geometry](#)
 - [Geometry Download](#)
- [About this document ...](#)

- Geometry available now on the wiki
- Available as **html**'ed latex or **pdf**
- Combination of user documentation and overview developer documentation
 - Intended to be read with doxygen for details

Chris 2012-06-13

MAUS-Status.odp - Libr... maus_user_guide - Mo...

Documentation (cont)



Project management
Rogers

Build system
Rogers

Geometry + fields
Rogers/Littlefield

TOF
Rajaram

Data Unpacking
Karadzhov

Detector Integration
Rogers/Lane

QA
Rogers

Geant4 Simulation
Rogers

Tracker
Dobbs/Santos et al

EMR
Karadzhov/Ruslan

Accelerator physics
analysis
Rogers/Lane

Documentation
Rogers

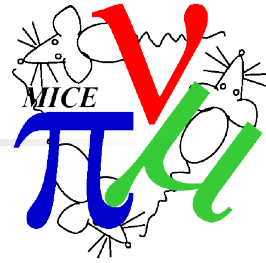
Data flow/API
Rogers/Richards

Ckov
Cremaldi/Kafka

KL
Bogomilov

- Documentation exists for tracker but is not integrated
- Documentation exists for data structure but API and core docs are missing

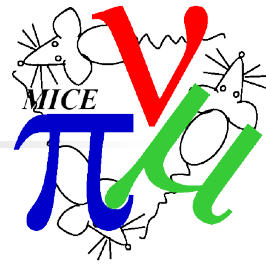
Python Test Coverage



Module name	Code Coverage	Owner
ReducePyTOFPlot	4%	Rajaram
MapPyTOFPlot	12%	Rajaram
framework.merge_output	32%	Richards (legacy from Jackson)
framework.input_transform	33%	Richards (legacy from Jackson)
MapPyScalersDump	45%	Karadzhov
ReducePyCkov	50%	Kafka
docstore.MongoDBDocumentStore	65%	Richards (legacy from Jackson)
docstore.DocumentStore	68%	Richards (legacy from Jackson)
OutputPyJSON	82%	
framework.utilities	83%	
Go	88%	
InputPyJSON	89%	

- Coverage is “% of lines executed by the unit tests”

Cpp Test Coverage



SciFi recon tests never ran...

(Rajaram, Karadzhov
Kafka)

Dobbs

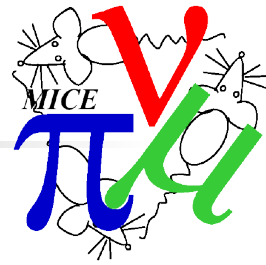
Richards

Karadzhov, Rajaram
Karadzhov

Rogers

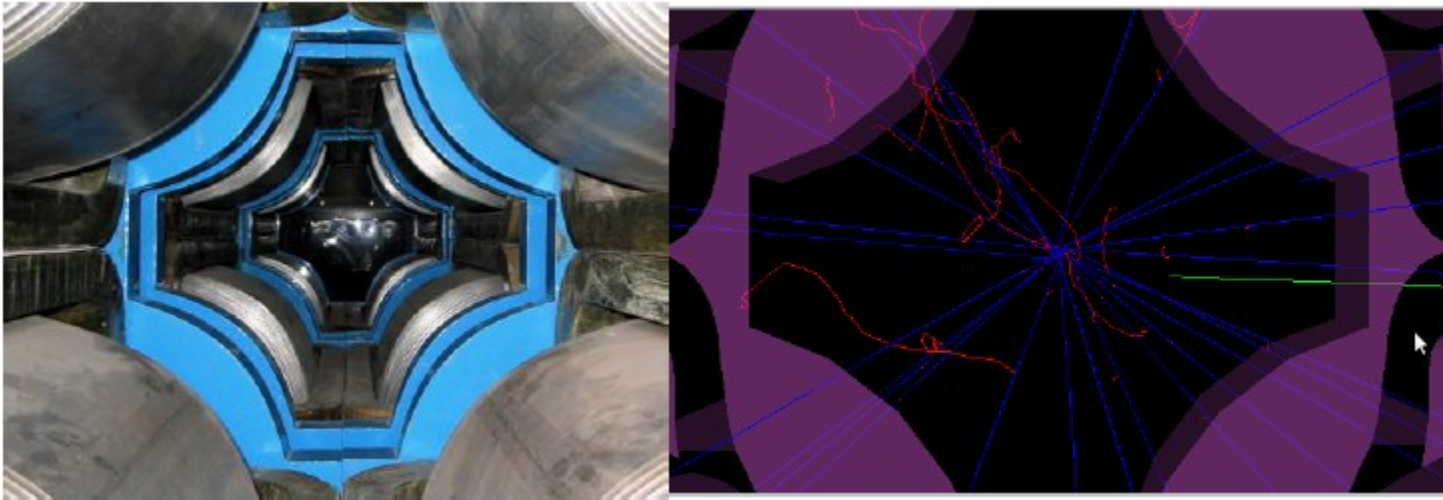
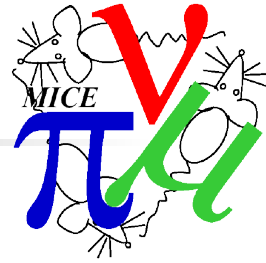
Directory	Line Coverage ↕
src/common_cpp/DataStructure	50.2 % 1282 / 2554
src/common_cpp/DetModel	100.0 % 12 / 12
src/common_cpp/DetModel/SciFi	33.3 % 45 / 135
src/common_cpp/FieldTools	94.0 % 328 / 349
src/common_cpp/JsonCppProcessors	88.0 % 662 / 752
src/common_cpp/JsonCppStreamer	51.9 % 139 / 268
src/common_cpp/Maths	89.2 % 1606 / 1800
src/common_cpp/Optics	93.0 % 160 / 172
src/common_cpp/Recon/SciFi	87.6 % 557 / 636
src/common_cpp/Simulation	92.1 % 877 / 952
src/common_cpp/Utils	68.5 % 536 / 783
src/input/InputCppDAQData	64.0 % 369 / 577
src/input/InputCppDAQOfflineData	75.0 % 30 / 40
src/input/InputCppRoot	69.1 % 38 / 55
src/map/MapCppPrint	90.5 % 19 / 21
src/map/MapCppSimulation	96.2 % 51 / 53
src/map/MapCppTOFDigits	80.9 % 157 / 194
src/map/MapCppTOFMCDigitizer	84.8 % 224 / 264
src/map/MapCppTOFSlabHits	93.5 % 115 / 123
src/map/MapCppTOFSpacePoints	90.3 % 149 / 165
src/map/MapCppTrackerMCDigitization	80.1 % 149 / 186
src/map/MapCppTrackerRecon	75.6 % 124 / 164
src/output/OutputCppRoot	82.7 % 43 / 52
src/reduce/ReduceCppTracker	0.5 % 1 / 187

Code Infrastructure



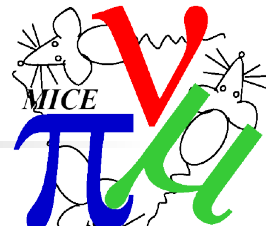
- Online
 - We ran in March and again in May
 - Software worked okay - but needed SOC
 - Some UI issues that need to be worked through
- Data structure + API
 - We have revised the data structure to include a particle event in the reconstruction
 - We now have a hard coded, documented data structure
 - Output in ROOT format or JSON
- Plans
 - Interface classes coming soon to hard code the existing API (Alex Richards)
 - Run-by-run control and global handling coming soon

Geometry (Matt Littlefield)

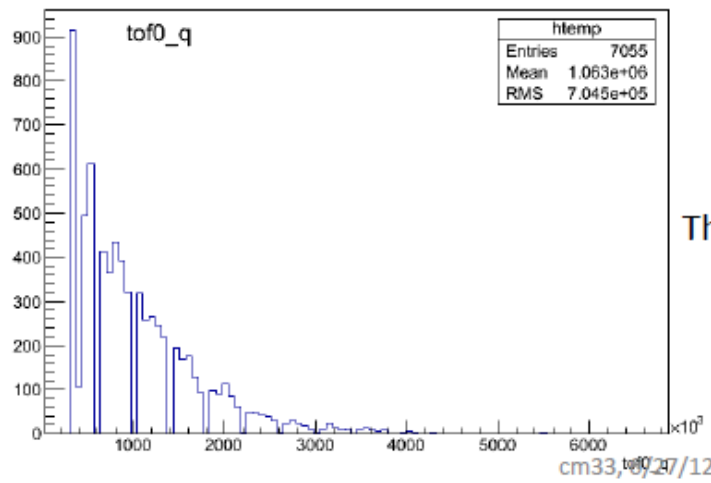
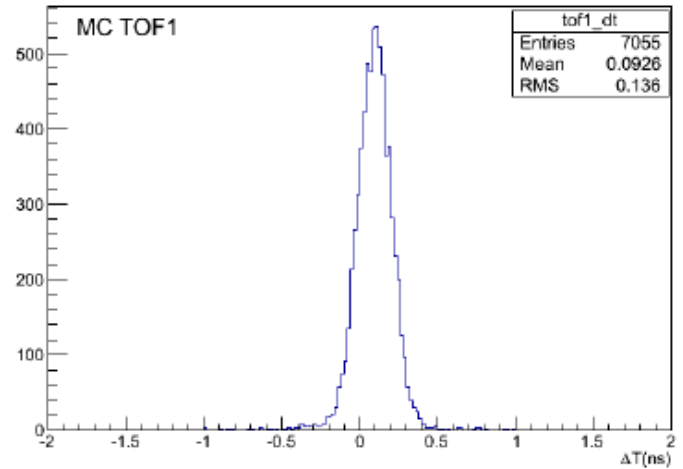
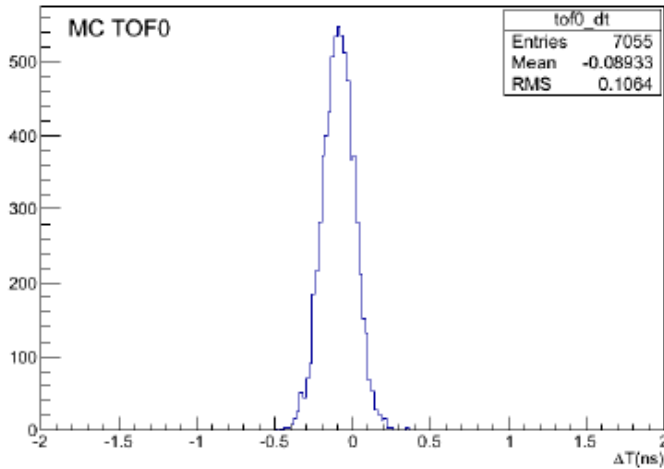


- MICE Step I geometry is now implemented in CAD and interfaced to MAUS
- Some integration issues need to be resolved
 - Needs a look from a physics point of view
- Still risk regarding processing time

TOF (Durga Rajaram)

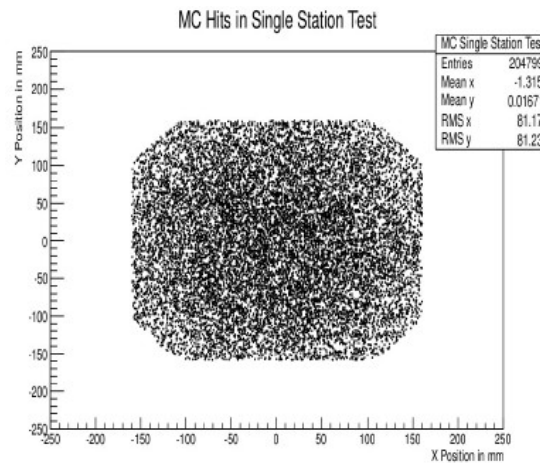
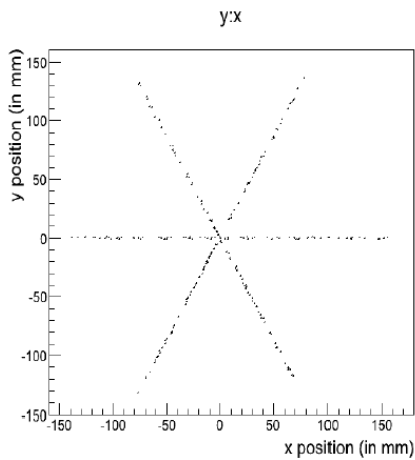
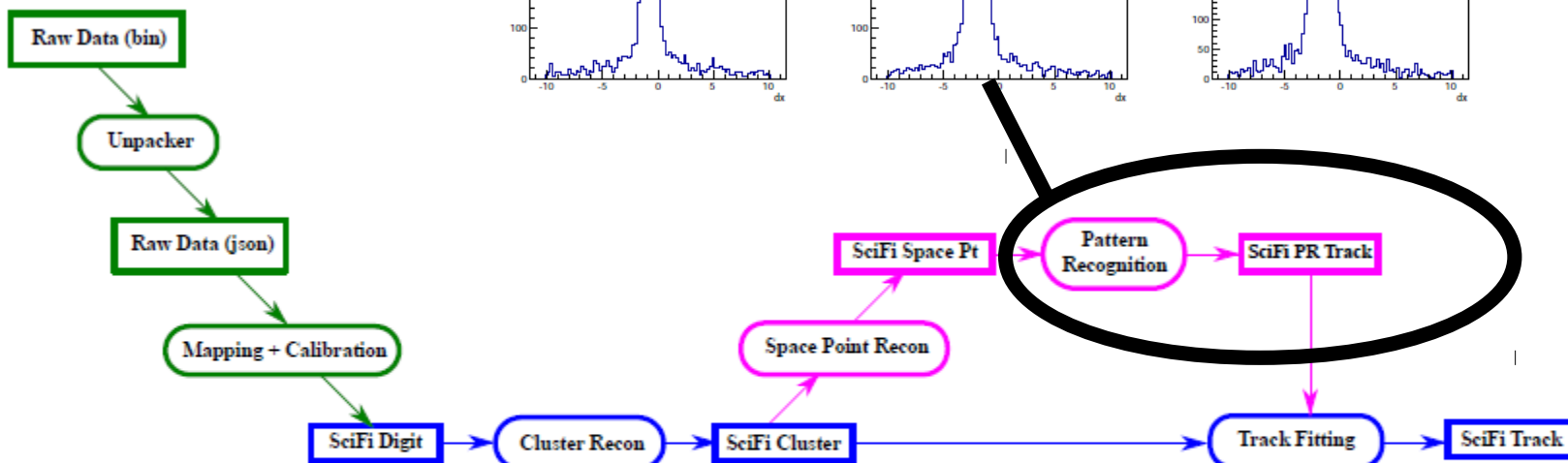
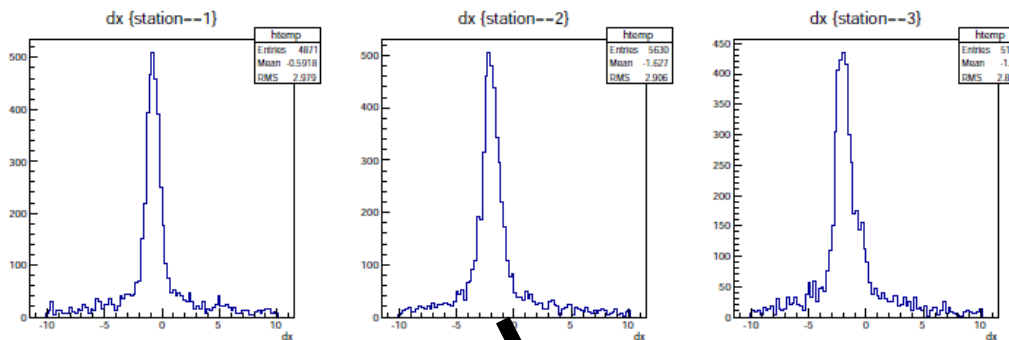
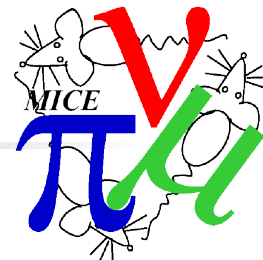


- Resolutions are reasonable as a starting point, but the smearing could probably be tweaked

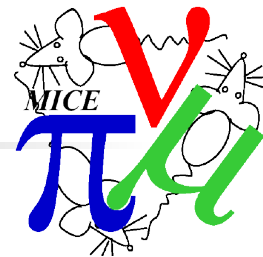


The charge digitization is obviously messed up

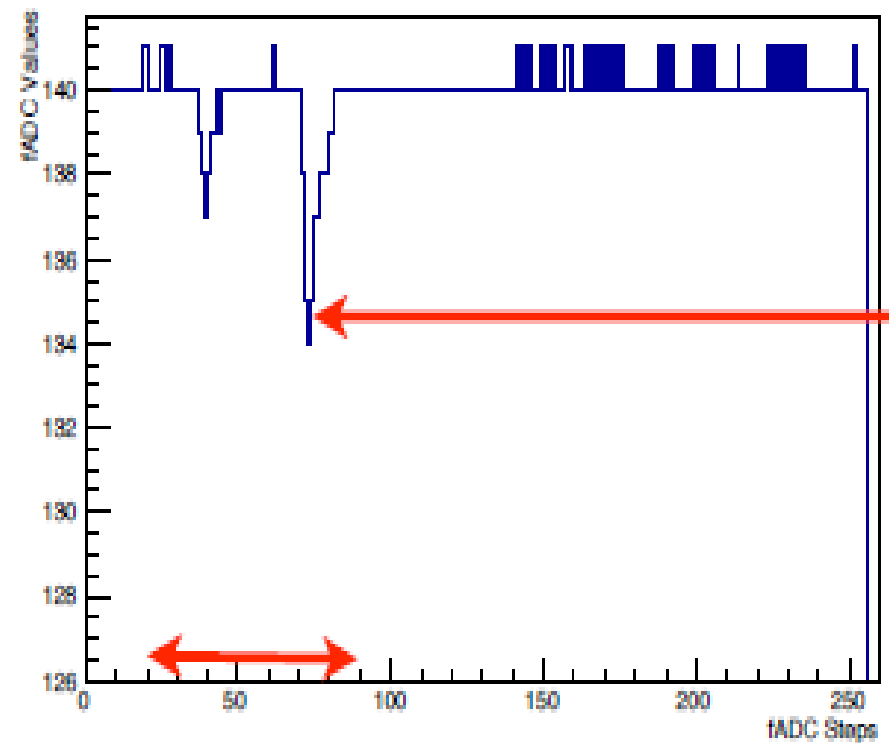
Tracker (Chris Heidt, Adam Dobbs)



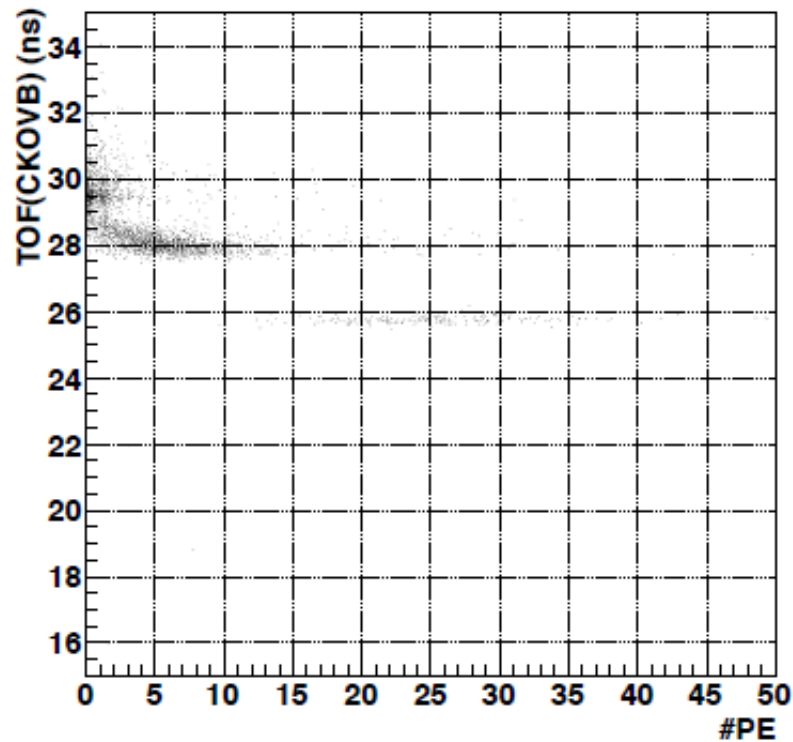
Ckov (Gene Kafka)

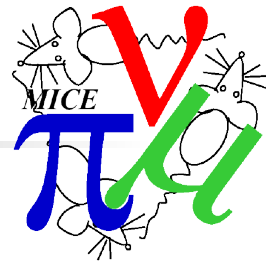


fADC Spectrum at Channel 0



tof_B





Let me point out few characteristics of existing KL code:

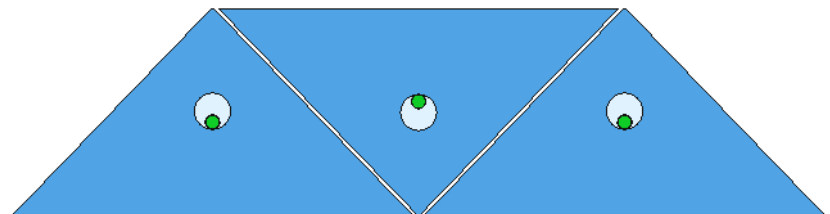
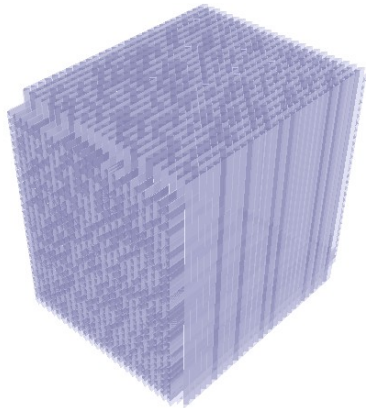
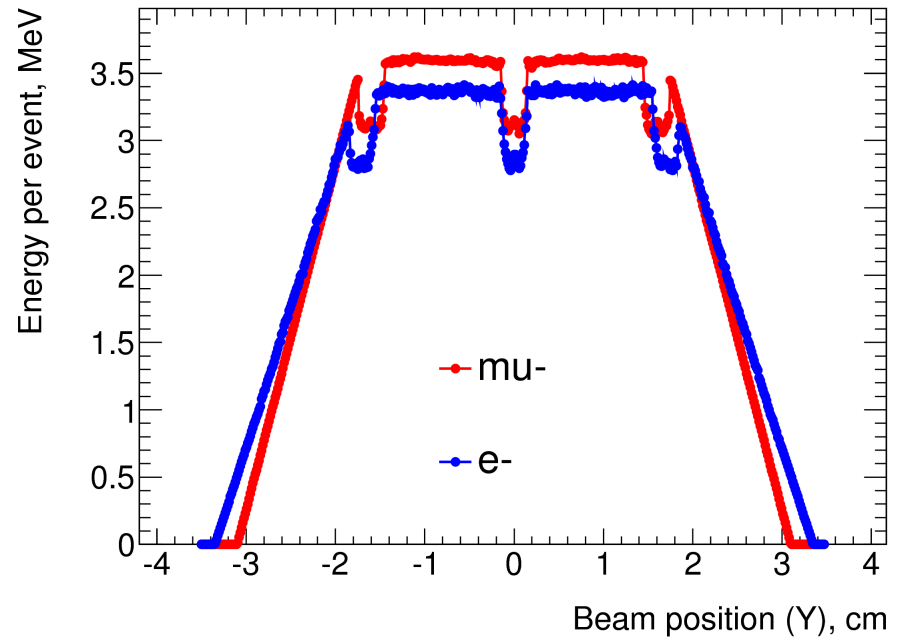
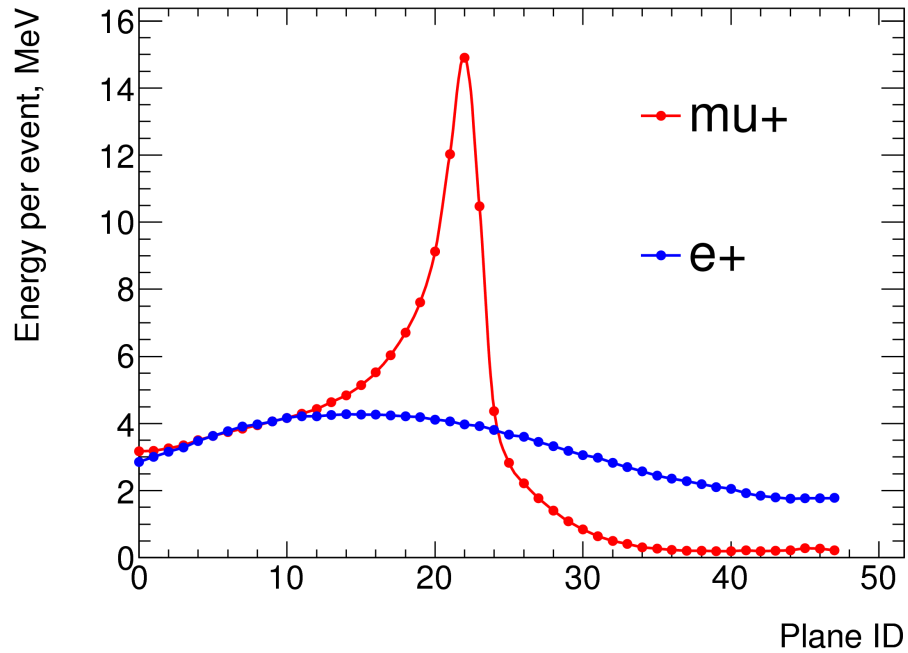
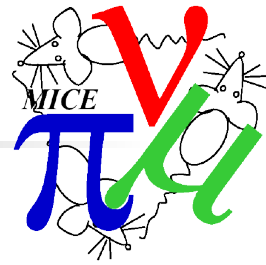
- 1) data structure is spill-by-spill
- 2) data are represented in json format only
- 3) it isn't merge with the trunk

But times are changing, philosophy is changing and MAUS data structure is changing too...

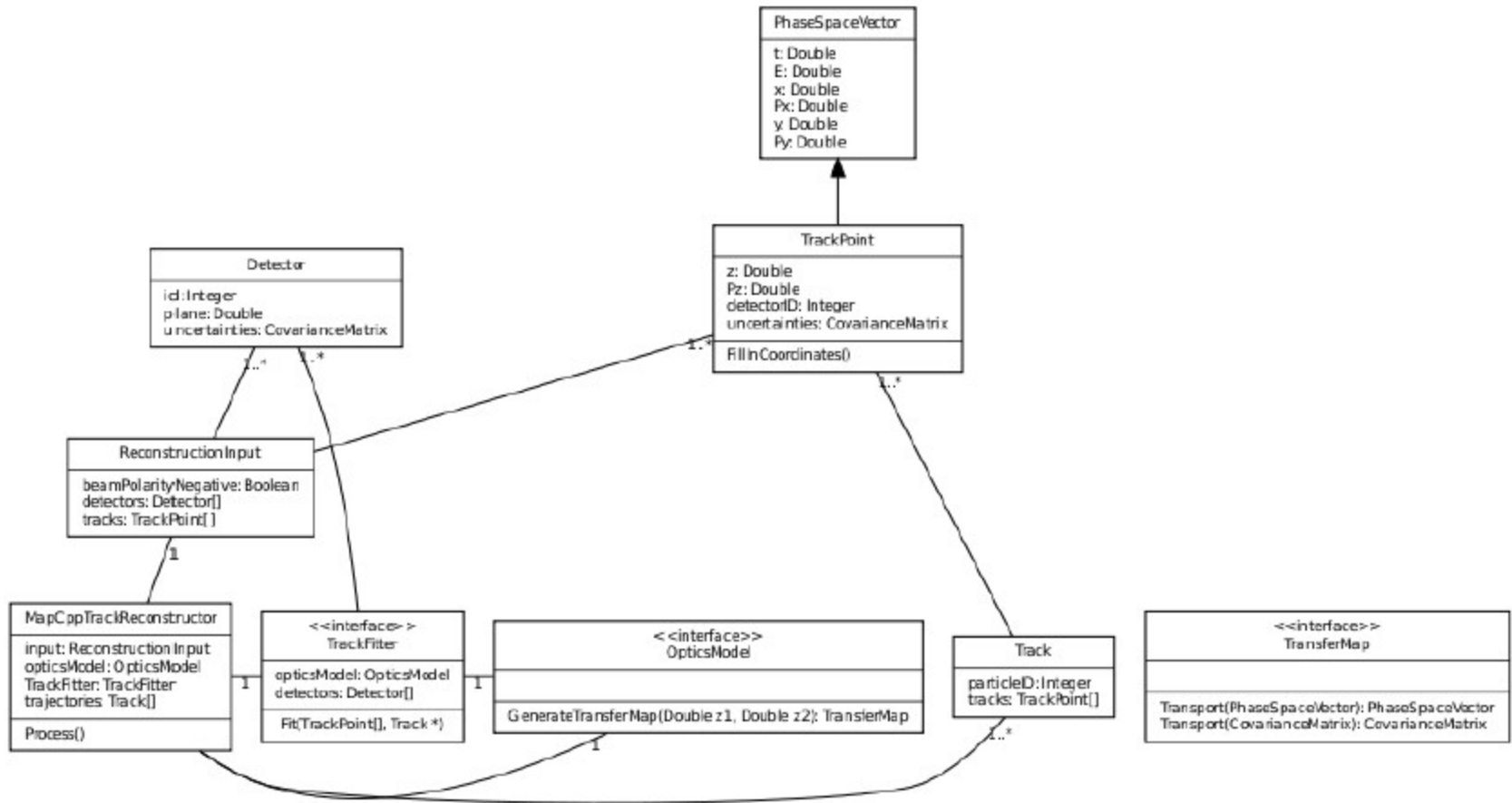
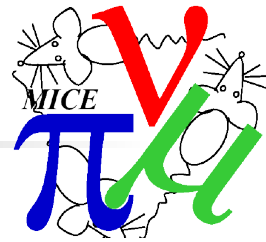
In order to fit KL with the new wave, I am doing

- 1) changing data structure from spill-by-spill to event-by-event
- 2) to represented data not only in json but in root format too
- 3) to merge KL code with the trunk in a week

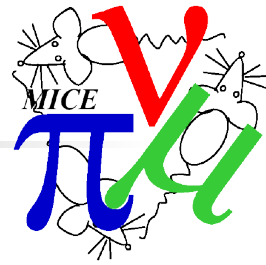
EMR (Leila Haegel)



Global Reconstruction (Peter Lane)



Risk and Schedule



- Tracker looks like it is on schedule for April
- TOF is okay
- EMR and Ckov are at-risk
 - But impact is lower
- Main risk to having working recon for Step IV comes from global reconstruction
 - Probably we review at the next CM and then think about getting more help on this
 - (6 month timeline is about right to get something done)
- Geometry is also a risk
 - Nervous about processing time for tracking
 - Needs a look from a more “physics” point of view
 - Presumably comes with implementation of Step IV geometry, Step I analyses