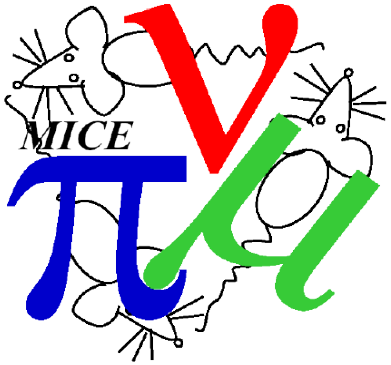
MAUS Status and Plans

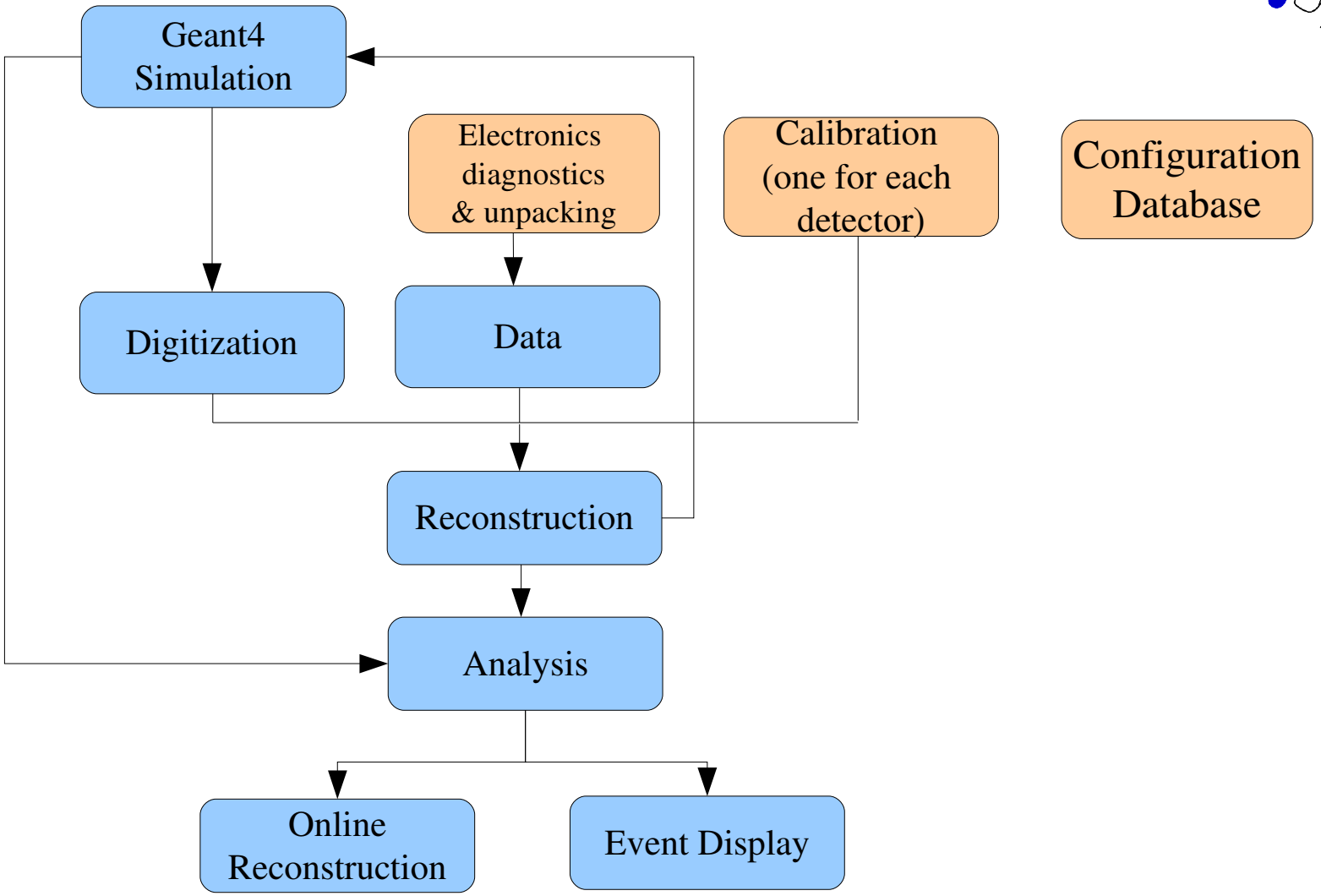


Chris Rogers,
ASTeC,
Rutherford Appleton Laboratory

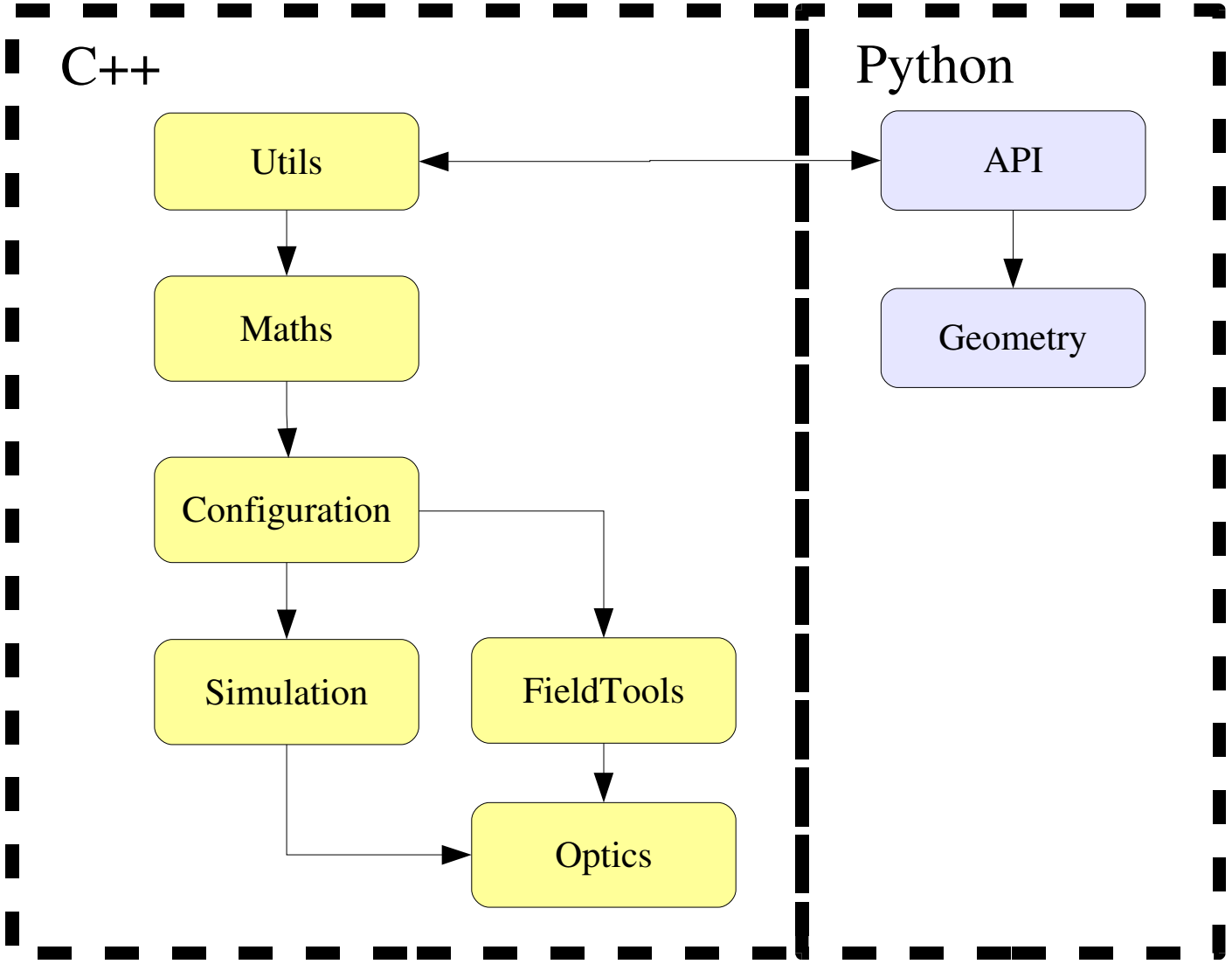
MAUS

- Project overview
- Test coverage
- Detector Integration package
- Software workshop
- New Features
- Progress against schedule
- Aims for next time

Data Flow


Geant4 Simulation
Electronics diagnostics & unpacking
Calibration (one for each detector)
Configuration Database
Digitization
Data
Reconstruction
Analysis
Online Reconstruction
Event Display

Common Code


C++
Python
Utils
API
Maths
Geometry
Configuration
Simulation
FieldTools
Optics

Responsibility/Block Diagram

Project management
Rogers

Build system
Tunnell

QA
Rogers

Documentation
Rogers

Geometry + fields
Rogers

Geant4 Simulation
Rogers

Data flow/API
Rogers

TOF
Karadzhov

Tracker
Dobbs

Ckov
Cremaldi

KL
Bogomilov?

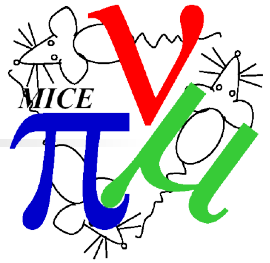
Data Unpacking
Karadzhov

EMR
Karadzhov

Detector Integration
Analysis group

Accelerator physics analysis
Analysis group (Rogers?)

Test Coverage



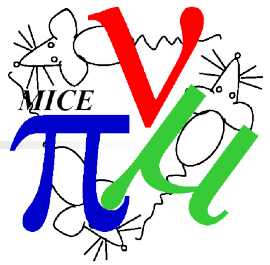
- Test coverage measures what proportion of code is tested

```
def is_positive(x):          # line 1
    if x > 0:                #      2
        return True          #      3
    if x < 0:                #      4
        return False         #      5

def my_test():
    assertTrue(is_positive(1))
```

- Consider test above
  - Test never checks that we return False for a negative number
  - Line coverage is 50% - (lines 2,3 are tested; lines 4,5 are not)

Test Coverage



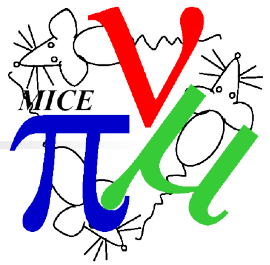
- Test coverage measures what proportion of code is tested

```
def is_positive(x):          # line 1
    if x > 0:                #      2
        return True          #      3
    if x < 0:                #      4
        return False         #      5

def my_test():
    assertTrue(is_positive(1))
    assertTrue(is_positive(-1))
```

- Consider test above
  - Test never checks that we return False for a negative number
  - Line coverage is 50% - (lines 2,3 are tested; lines 4,5 are not)
- Now line coverage is 100%
  - But there is still a bug! What if x is 0?
  - Branch coverage tells us that we didn't test all possible options

Test Coverage



- Test coverage measures what proportion of code is tested

```
def is_positive(x):          # line 1
    if x > 0:                #      2
        return True          #      3
    if x < 0:                #      4
        return False         #      5

def my_test():
    assertTrue(is_positive(1))
    assertTrue(is_positive(-1))
    assertTrue(is_positive(0))
```

- Consider test above
  - Test never checks that we return False for a negative number
  - Line coverage is 50% - (lines 2,3 are tested; lines 4,5 are not)
- Now line coverage is 100%
  - But there is still a bug! What if x is 0?
  - Branch coverage tells us that we didn't test all possible options
- Aim for line coverage >~ 90%

Test Coverage - C++

	Line Coverage		Branches	
src/common_cpp/DetModel	91.7 %	11 / 12	33.3 %	4 / 12
src/common_cpp/DetModel/SciFi	33.6 %	43 / 128	16.0 %	8 / 50
src/common_cpp/Simulation	91.4 %	804 / 880	52.9 %	257 / 486
src/common_cpp/Utils	69.8 %	448 / 642	45.7 %	216 / 473
src/input/InputCppDAQData	73.0 %	348 / 477	46.5 %	127 / 273
src/legacy/BeamTools	36.0 %	942 / 2617	26.4 %	408 / 1543
src/legacy/Config	35.7 %	730 / 2047	29.9 %	411 / 1376
src/legacy/DetModel/Ckov	1.2 %	2 / 169	5.1 %	4 / 78
src/legacy/DetModel/EMR	9.1 %	1 / 11	20.0 %	2 / 10
src/legacy/DetModel/KL	2.2 %	3 / 136	4.8 %	6 / 126
src/legacy/DetModel/TOF	2.6 %	1 / 39	20.0 %	2 / 10
src/legacy/DetModel/Virtual	45.7 %	64 / 140	25.0 %	21 / 84
src/legacy/EngModel	28.4 %	95 / 335	24.3 %	68 / 280
src/legacy/Interface	45.8 %	2159 / 4713	39.5 %	1146 / 2900
src/legacy/Optics	1.3 %	29 / 2305	2.9 %	32 / 1099
src/legacy/Simulation	58.3 %	354 / 607	26.8 %	81 / 302
src/map/MapCppPrint	95.2 %	20 / 21	50.0 %	3 / 6
src/map/MapCppSimulation	92.0 %	46 / 50	75.0 %	6 / 8
src/map/MapCppTOFDigitization	65.4 %	89 / 136	37.3 %	62 / 166
src/map/MapCppTOFDigits	85.0 %	136 / 160	68.3 %	56 / 82
src/map/MapCppTOFSlabHits	92.6 %	112 / 121	62.0 %	31 / 50
src/map/MapCppTOFSpacePoints	90.2 %	138 / 153	53.2 %	58 / 109
src/map/MapCppTrackerDigitization	87.2 %	163 / 187	58.3 %	119 / 204

Test Coverage - Python

Name	Stmts	Miss	Line Coverage
Configuration	77	0	100.00%
ErrorHandler	62	1	98.00%
Go	93	5	95.00%
InputPyEmptyDocument	18	0	100.00%
InputPyJSON	23	0	100.00%
InputPySpillGenerator	24	0	100.00%
MapPyBeamMaker	87	1	99.00%
MapPyDoNothing	12	0	100.00%
MapPyFakeTestSimulation	19	0	100.00%
MapPyGroup	43	33	23.00%
MapPyPrint	20	0	100.00%
MapPyRemoveTracks	37	0	100.00%
MapPyValidateSpill	21	10	52.00%
OutputPyDoNothing	7	0	100.00%
OutputPyImage	50	0	100.00%
OutputPyJSON	34	6	82.00%
OutputPyRootHistogram	31	22	29.00%
ReducePyDoNothing	9	0	100.00%
ReducePyHistogramTDCADCCounts	49	0	100.00%
ReducePyMatplotlibHistogram	72	0	100.00%
SchemaSchema	1	0	100.00%
SpillSchema	17	0	100.00%
beam	167	1	99.00%

Documentation

- Up to now focus has really been on improving code quality and testing
  - We need to turn our attention to documentation or it will bite us down the road
  - Burden on Rogers to provide the infrastructure
    - Some documentation skeleton in (latex)
    - Needs some example entries to help people get started/check that the skeleton makes sense
    - Conversion of existing (legacy) documentation into this framework

Detector Integration Task (1)

- Two things have happened
  - Analysis group ownership has changed
  - Chris Tunnell has asked for someone else to take responsibility for detector integration task
- Somehow we need to get the detectors to talk to each other
  - Track fitting/extrapolation between detectors
  - PID
  - MICE Event
- The code for this belongs in MAUS
  - We want to run it in the control room
- But the task belongs to analysis group

Detector Integration Task (2)

- Work progress
  - Software group have started work on transfer matrix
    - Track/error extrapolation through arbitrary EM field + materials
    - E.g. for magnet or detector vs detector alignment study
  - Software group delivers
    - Space points in TOF
    - Tracks from tracker
    - EMR energy deposition
    - Ckov light yield
- Particle event reconstruction is entirely software responsibility
  - Sort electronics signal by the set of triggers
  - Feed into data quality check
  - Not existing in current code
- Online reconstruction/data quality checking is entirely software responsibility

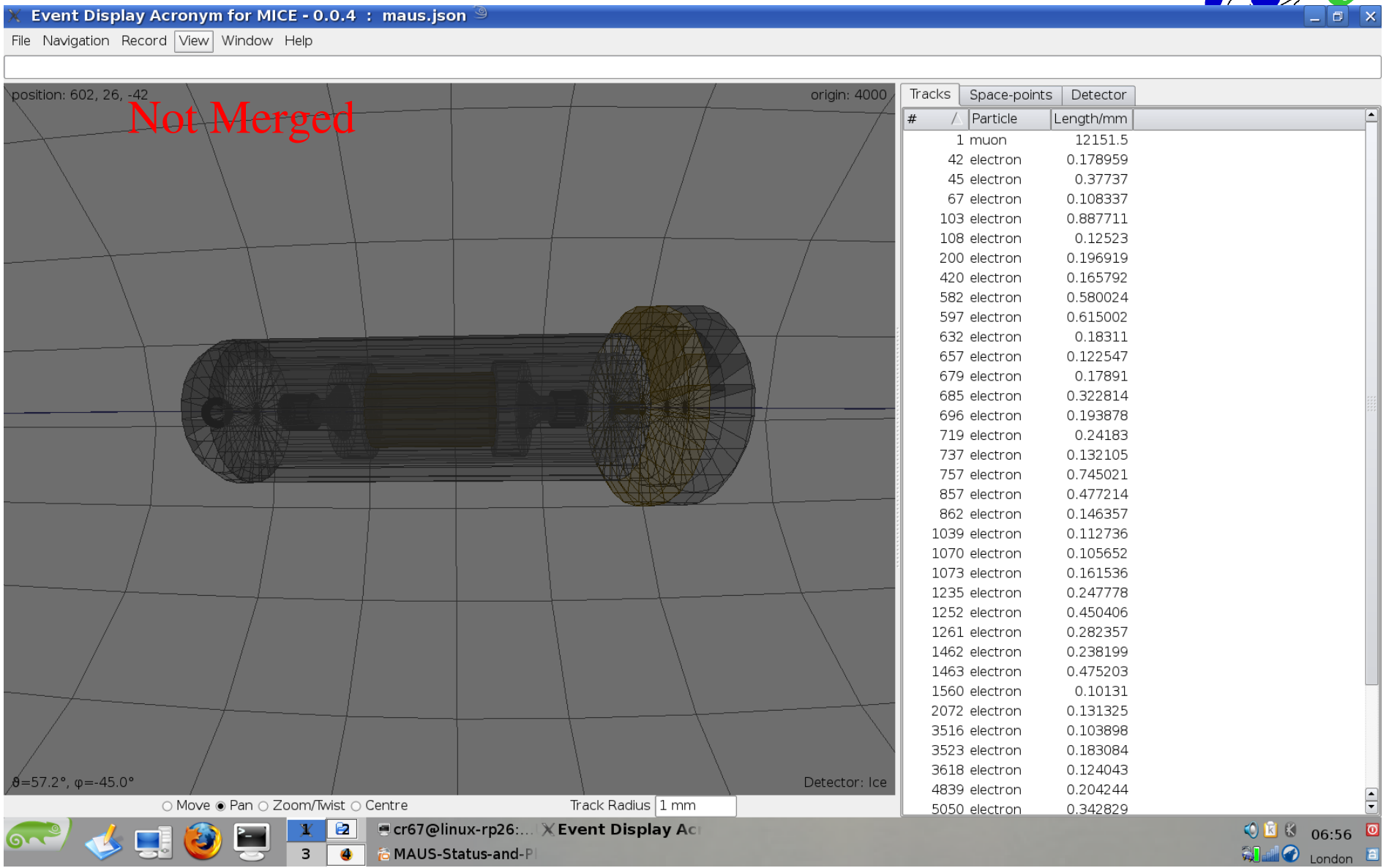
New Features

- Beam generation
- Json browser (ntuple browser)
- Data unpacking (Yordan Karadzhov)
- TOF reconstruction to space points (Yordan Karadzhov)
- Tracker Monte Carlo (Edward Santos)
- CAD Geometry Import (Matt Littlefield)
- Online + Offline detector plots (Chris Tunnell + Mike Jackson)
- 3D Visualisation (Matt Robinson)

Beam Generation

- New beam generation code
  - Generate reasonable spill time structure
  - Generate Gaussian multivariate beams with various beta functions etc
  - Mix pions, muons, electrons in a reasonable way
  - Pull beams from binomial distribution

3D Visualisation (Matt Robinson)



Software Workshop

- Successful workshop
  - Great effort by Linda Coney, Chris Tunnell for local coordination
  - More from Chris Tunnell

Progress against schedule

Item	Responsible	Date	Revised Date
0.1.0			
Simulation refactor	Rogers	01/10/11	**01/01/12**
Unpacking PID detectors	Karadzhov	01/10/11	**04/10/11**
Unpacking tests	Karadzhov	01/10/11	**01/01/12***
TOF spacepoints	Karadzhov	01/10/11	**21/10/11**
Tracker MC	Dobbs	?	**26/10/11**
CAD import algorithms	Littlefield	01/10/11	**01/11/11**
Visualisation Prototype	Robinson	01/10/11	**01/11/11**
0.2.0			
Prototype online histos	Tunnell	01/11/11	21/10/11

- I didn't make a 0.1.0 because I wanted the “simulation” work to go in
- Probably need to push a 0.1.0 in any case
- Caveat on schedule:
  - Not resource loaded
  - No dependency analysis
  - Not robust

Slip

- Simulation
  - The goal here is to have a well tested and well documented module
    - Exercise work flow surrounding test suites and documentation
    - E.g. add physics validation tests, load tests
    - Fit in a different area to the established unit tests
  - Prioritisation goes to work that needs doing for November running
    - Slip this work to make sure we have e.g. TOF and unpacking code
- Unpacking
  - Unpacking was blocking so first draft has been merged with trunk
    - Blocks much of the detector reconstruction code
  - Testing is substandard and needs improvement

Aims for Next Time

0.2.0			
Unpacking tests	Karadzhov	01/10/11	**01/01/12***
Simulation refactor	Rogers	01/10/11	**01/01/12**
TOF tracks	Karadzhov	01/01/12	01/01/12
Prototype Ckov recon	Cremaldi	01/01/12	01/01/12
Online histos finished	Tunnell	01/01/12	01/01/12
Visualisation UI finalised	Robinson	01/01/12	01/01/12

- Make a release first thing in the new year
- General aim is to test full workflow in February running
  - Online analysis
  - “Official” production monte carlo of experiment
  - “Official” offline reconstruction of data
- Beamline monte carlo missing
- TOF/trigger digitisation missing

Summary

- Since last time
  - 106 issues closed
  - 49 issues opened
  - 100 issues remain open
- Successful software workshop
- Change in working mode has been a stressful time
  - A lot to learn for even experienced developers
  - Still need to work on high level testing
    - Physics validation, load testing, ...
  - Still need to work on documentation
  - Schedule is still in development
    - 1 year into the job, this is late
- New developers are starting to really become productive
- We have a lot of progress in 4 months
  - Great effort from all involved
  - Really great improvement in quality of code produced
  - Huge amount to do