MAUS - Feature #1411

KL MC

07 February 2014 06:38 - Rajaram, Durga

Status:	Open	**Start date:**	07 February 2014
Priority:	Normal	**Due date:**	
Assignee:	Bogomilov, Mariyan	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Future MAUS release		
Workflow:	New Issue		

Description

KL MC

Need simulation of KL detector in MAUS.

Geometry

- description exists in legacy geometry
- are the material definitions and internal geometries correct?

Hits

- there is some legacy code for the sensitive detector in src/legacy/DetModel/KL/ -- needs to be checked/modified
- add KLChannelId and typedef KLHit [src/common_cpp/DataStructure]
- add KLChannelIdProcessor [src/common_cpp/JsonCppProcessors]

Digitization

- digitize GEANT hits - add src/map/KLMCDigitizer
- check data structure [src/common_cpp/DataStructure/KLDigit.xx]

Reconstruction

- reconstruction code exists
- the reconstruction [MapCppKLCellHit] should treat MC digits same way as it does digits from real data

Tests

- unit tests
- integration tests to validate simulation+reconstruction chain

Documentation

- add TeX documentation

History

#1 - 11 March 2014 15:38 - Rajaram, Durga

- File simulate_kl.log added

The KL digitizer crashes on some events:

```
python bin/user/simulate_kl.py --simulation_geometry_filename=Stage4.dat
```

Full log attached.

```
..
#6  0x00007fe9b0882345 in MAUS::MapCppKLCellHits::fillCellHit(Json::Value, Json::Value) () from /home/durga/tr
```

```
unk-debug/build/_MapCppKLCellHits.so
#7  0x00007fe9b08831cc in MAUS::MapCppKLCellHits::makeCellHits(Json::Value) () from /home/durga/trunk-debug/bu
ild/_MapCppKLCellHits.so
#8  0x00007fe9b0883ac6 in MAUS::MapCppKLCellHits::process(std::basic_string<char, std::char_traits<char>, std:
:allocator<char> >) () from /home/durga/trunk-debug/build/_MapCppKLCellHits.so
#9  0x00007fe9b0888d77 in _wrap_MapCppKLCellHits_process () from /home/durga/trunk-debug/build/_MapCppKLCellHi
ts.so
#10 0x00007fe9c92aa894 in ext_do_call (f=<value optimized out>, throwflag=<value optimized out>) at Python/cev
al.c:4323
#11 PyEval_EvalFrameEx (f=<value optimized out>, throwflag=<value optimized out>) at Python/ceval.c:2705
..
```

#2 - 19 March 2014 05:17 - Rajaram, Durga

Mariyan,

The crash is happening because of a division by zero in MapCppKLCellHits, line: 210-211

```
xDocCellHit["charge_product"] = 2 * xChargeDigit0 * xChargeDigit1 / (xChargeDigit0 + xChargeDigit1);
```

I fixed this by setting the charge_product to 0 if (xChargeDigit0 + xChargeDigit1) == 0
I'll push that if you think that's reasonable.

The question though is why the MC digitizer is giving some digits of 0 -- very low energy deposit+smearing round down?

#3 - 19 March 2014 11:33 - Bogomilov, Mariyan

Thanks Durga,

please push it. I was not able to investigate it until now. I'll try to see what is going on in details.

#4 - 19 March 2014 15:39 - Rajaram, Durga

- File kl_mcreco.png added

- File kl_pie.png added

Ok great, thanks!
FYI, if you find it useful, I have attached some plots from naive quick simulations I ran

Files

simulate_kl.log	26.1 KB	11 March 2014	Rajaram, Durga
kl_mcreco.png	12.6 KB	19 March 2014	Rajaram, Durga
kl_pie.png	9.1 KB	19 March 2014	Rajaram, Durga