

# MAUS - Feature #1182

## Trigger Digitisation

03 December 2012 16:35 - Rogers, Chris

<b>Status:</b>	Open	<b>Start date:</b>	03 December 2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Bayes, Ryan	<b>% Done:</b>	0%
<b>Category:</b>	Simulation	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Future MAUS release		
<b>Workflow:</b>	New Issue		

### Description

Currently, digits are associated together by assuming that all digits from a given primary particle lead to the same MC event. This does not enable us to properly MC simultaneous events and noise. Need a new mapper that is responsible for trigger digitisation, say MapMCDigitisation that inherits from API/MapBase. Algorithm is something like:

- Reconstruction proceeds as previously to the end of individual detector digitisation.
- In MapMCDigitisation::birth() perform a datacard lookup that decides which detector(s) is the trigger detector. DAQ allows one of TOF0,1,2 or AND combinations
- In MapMCDigitisation::process() loop over TOF digits. Search for 2 digits in a TOF station to make a trigger.
- In MapMCDigitisation::process() loop over all digits. Make a new reconstructed event list, with one reconstructed event for each trigger. Replace the reconstructed event (one event per primary) with the modified reconstructed event.

### History

#### #1 - 22 February 2013 16:10 - Bayes, Ryan

- File *MCTriggerStructure.pdf* added

Some refinements to the above considerations that resulted from the Software workshop following CM35. Some changes to the general data structure will have to be made including the addition of a new branch to handle the timing information and coordinate between the MCEventArray and RecEventArray branches. Further description appear in the attached file.

#### #2 - 22 February 2013 16:12 - Rogers, Chris

Added Durga as a watcher... he might want to receive the expanded specification document.

#### #3 - 22 May 2013 11:57 - Rogers, Chris

I had a look at your code - I didn't really get into it before noticing that

- there are no comments
- there are no tests

I can't review code without knowing what it is supposed to do. Please see the section [Development Workflow](#)

#### #4 - 16 June 2013 18:21 - Bayes, Ryan

Test and comments now added.

Discussions with Durga raised a potential issue with the usage of the TOFdigit leading time as the trigger time. The leading time includes corrections for the propagation time, which we want in the trigger, and a time walk correction, which includes an ADC integration time that may exceed the length of the trigger window. Thus the time walk correction should probably be removed from the trigger time. This is probably as simple as redefining the "leading time" in the MC TOF digits as just the true MC time corrected by the propagation time and correcting the digit times by the time walk correction within the MC Trigger Digitization before the TOF digits are pushed to the reconstruction branch. Another possibility is adding a "raw leading time" to the TOF MC digits.

#### #5 - 16 June 2013 22:50 - Rogers, Chris

- Assignee changed from Bayes, Ryan to Rogers, Chris

Okay, will have a look.

#### #6 - 27 June 2013 16:18 - Rogers, Chris

I had a look at r687 in `bzr+ssh://bazaar.launchpad.net/~ryan-bayes/maus/devel/` and a few questions presented:

Datastructure is a user interface and documentation should be standalone, contained within doxygen output:

- What is MCTrigger? What does it represent?
- What is MCTriggerInfo::\_time? Is this the time that the trigger came in relative to some 0? Some offset between tof and trigger? something else?
- What is MCTriggerInfo::\_event? Is this the spill? The MC particle event index? The recon particle event index? Something else?

MapCppMCTriggerDigi:

- `_triggerConfig`, `_windowlength`, `_trailingtime` - I guess this comes from CDB for MC of reconstructed event; or datacards for pure MC run. Could you make this a datacard for now; Durga, do you know if this stuff is in the CDB? If not, it should be (but that is a long road to tread).
- comments: could you put comments in the header file, as well as/instead of the cpp file? The header file goes into doxygen, the cpp file does not. I would like for the purpose and main functionality of the class to be documented in doxygen. I do like the comments in the cpp file though, nice and detailed.

Tests:

- Well, your line coverage is 50% and your test is, to be blunt, crap. Without thinking too hard, I would expect test cases like:
  - multiple hits in the same trigger
  - no hits in the trigger window
  - hits partially in the trigger window
  - trigger is TOF0/TOF1/TOF2
  - etc, you know the code better than me
- You should be checking that the return time offset etc is as expected

Maybe you forgot to push or I am looking at the wrong branch?

#### #7 - 03 July 2013 11:40 - Bayes, Ryan

Updated the comments in the header files. Also expanded the unit test to include cases of multiple particle pileups and empty events.

I need to verify the use of the primary particle reference time (`spill["mc_events"][ievent]["primary"]["time"]`) for pileup. The resulting trigger times look very odd in context and I think a better reference time for a pileup event would be useful.

#### #8 - 03 July 2013 13:50 - Rogers, Chris

- Assignee changed from Rogers, Chris to Bayes, Ryan

I'm not quite sure what you mean by this. Are you asking for advice? If so could you be more specific?

#### #9 - 04 July 2013 12:16 - Bayes, Ryan

Turned out that the main problem was one of definitions --- I needed to redefine the trigger time scale for pileup to look like the single particle trigger. I also had to put everything in TDC units and make sure that all of the digits in a trigger window do indeed get written to the data structure; a mistake in the loop structure meant that I was unintentionally overwriting TOF hit information from one particle to the next in the case of multiple particle pileup.

The code is now ready for a merge.

#### #10 - 28 August 2013 14:27 - Rogers, Chris

I fixed all of the style errors and checked that tests pass okay at `bash tests/style_tests.bash`. I haven't gone through in detail.

I had a query about the lines

```
spill["mctrigger_event"] = []

# check data integrity
if "mc_events" not in spill or type(spill["mc_events"]) != type([]):
    return json.dumps(spill)

# loop over all triggers and fill the recon event
for i, event in enumerate(spill["mctrigger_event"]):
    spill["mctrigger_event"].append({"triggeroffsets":{}})
```

in `src/map/MapPyMCTriggerEventSetup/MapPyMCTriggerEventSetup.py` line 61. The loop does nothing!

Should this really be

```
# loop over all triggers and fill the recon event
for i, event in enumerate(spill["mc_events"]):
    spill["mctrigger_event"].append({"triggeroffsets":{}})
```

or more "python-esque"

```
spill["mctrigger_event"] = [{"triggeroffsets":{}} for event in spill["mc_events"]]
```

I notice there is no test for this code.

Pushed to [bazaar.launchpad.net/~chris-rogers/maus/bayes-mc-trigger/](http://bazaar.launchpad.net/~chris-rogers/maus/bayes-mc-trigger/) and a test job set up at [http://test.mice.rl.ac.uk/job/MAUS\\_rogers\\_bayes\\_merge/](http://test.mice.rl.ac.uk/job/MAUS_rogers_bayes_merge/) against that branch.

**#11 - 28 August 2013 14:31 - Rogers, Chris**

- File `cpp_coverage.tar.gz` added

Attached cpp coverage report. Note that there is no coverage of any of the datastructure stuff. I think this should at least be exercised by the tests.

**#12 - 28 August 2013 14:33 - Rogers, Chris**

On python side I note (related to #10) `MapPyMCTriggerEventSetup` has only 33% line coverage

**#13 - 28 August 2013 16:00 - Rogers, Chris**

Something that I added - I made sure that all of the files had the usual MAUS GPL licence in

```
# This file is part of MAUS: http://micewww.pp.rl.ac.uk:8080/projects/maus
#
# MAUS is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# MAUS is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with MAUS. If not, see <http://www.gnu.org/licenses/>.
```

Ryan, can you confirm that you agree to this licence.

**#14 - 28 August 2013 16:10 - Bayes, Ryan**

Yes I agree. Thanks, Chris.

**#15 - 28 August 2013 18:07 - Rogers, Chris**

- Assignee changed from Bayes, Ryan to Rogers, Chris

Another bug - the `MCTriggers` are not included in the `SpillProcessor`. An integration test is needed I think to catch this sort of thing.

**#16 - 29 August 2013 14:19 - Rogers, Chris**

I have a working integration test. Two issues arising

1. When `triggerevent_pileup` is `True`, all events got pushed into the first MC. I don't **think** I screwed up the spill gate setup, but I will dig and debug.
2. The following item does not appear to have been implemented (or is not working)

In `MapMCDigitisation::process()` loop over all digits. Make a new reconstructed event list, with one reconstructed event for each trigger. Replace the reconstructed event (one event per primary) with the modified reconstructed event.

**#17 - 29 August 2013 15:01 - Bayes, Ryan**

Issue 1 doesn't sound right at all. Events should get pushed to the data structure, not the MC structure.

Second issue sounds like something is very broken and could be feeding back into issue 1 (i.e. related to the reason that things are writing to the MC structure rather than the data structure).

Is there anything I can do to help?

**#18 - 29 August 2013 15:15 - Rogers, Chris**

Ah, looks like it moves the digits but does not delete the (now empty) recon event.

**#19 - 29 August 2013 16:42 - Rogers, Chris**

Found a bug that may have been causing some of the confusion... in the process method

```
if (!_pileup) { // Only considering single particles as events
    if (triggerSingleParticle(ievent)) {
        correctDigitTimes(ievent);
        ^ should be tevent
        tevent++;
    }
    ievent++;
} else { // More complicated event topology
    correctDigitTimes(tevent);
    tevent++;
    ievent += _pileupoffset.size();
} else {
    ievent++;
}
}
```

#### #20 - 30 August 2013 16:57 - Rogers, Chris

Another bug - in triggerWithPileup(...) it looks like the do ... while loop only piles up adjacent hits. Need to do a sort by time stamp on the digits before entering this loop.

#### #21 - 30 August 2013 17:02 - Rogers, Chris

Oh, and I think we only need one of the lines

```
_pileupoffset.push_back(_triggertime);
```

and

```
_pileupoffset.push_back((_triggertime + primary_generationtime - current_generationtime));
```

Further, there is a potential memory problem in correctDigitTimes(...) where we loop over particlesInEvent::iterator particleIt and pileupoffset::iterator timeOffset within the same loop without explicitly checking for pileupoffset.end(). This has at least tripped me up and I think was a invalid read in the original code.

#### #22 - 28 November 2013 13:40 - Bayes, Ryan

- Assignee changed from Rogers, Chris to Bayes, Ryan

Rogers, Chris wrote:

Another bug - in triggerWithPileup(...) it looks like the do ... while loop only piles up adjacent hits. Need to do a sort by time stamp on the digits before entering this loop.

It is not adjacent hits that this loop acts upon, but hits in adjacent events. The contents of the loop asks if any hit in the event is contained by the time window. If it is then the event is number is recorded with the time offset relative to the trigger time. The problem is the same, however. If the events are not compiled in some kind of time order, then there will be a failure in the algorithm to identify mc events that participate in the time window. Of course that raises a question; why should the events be recorded outside of time order?

A solution is to create a vector of the event indices ordered in time, and remove entries from that vector as the events are added to a particular trigger window.

#### #23 - 28 November 2013 13:45 - Rogers, Chris

E.g. the MC beam generation algorithm just samples time from "some distribution". I can do a sort, but it would be a feature request. We should make sure that e.g. beam generator from G4BL that John is working from also does a sort-by-time. My feeling is that it would be better to do the sort in the trigger MC routine (as this is the one that needs it on a conceptual level).

#### #24 - 03 December 2013 16:24 - Bayes, Ryan

I have successfully introduced a time sort on the trigger events. The principle is that all events in a spill are considered as single event triggers and the leading tof0 time is extracted for the event. The leading tof0 time is then used to sort the resulting events in ascending order.

The code happily compiles after the trunk was merged into its branch, but something is not playing nice. The TOFDigits test is now broken; I am looking for the cause.

#### #25 - 08 January 2014 10:52 - Bayes, Ryan

I corrected some problems suggested by the unit tests for the mctrigger mapper, so the mctrigger map is ready to go with the corrections suggested by Chris.

I am still left with a minor mystery in my unit tests. If I run all the unit tests from using the "run\_tests.bash" script, the process is terminated at the "MapCppTOFDigits" test, with the following error

```
Test MapCppTOFDigits process method ... terminate called after throwing an instance of 'MAUS::Exception'
```

```
what(): Attempting to find Json property tof_key but not an object at JsonWrapper::GetPropertyStrict
(blah)/test/unit_test.bash: line 32: 1354 Aborted      python ${MAUS_THIRD_PARTY}/third_party/install/bin/nosetests -v
${MAUS_ROOT_DIR}/build
```

The test runs with no problem if I run the test in the "MapCppTOFDigits" directory.

#### #26 - 11 February 2014 12:44 - Bayes, Ryan

- Assignee changed from Bayes, Ryan to Rajaram, Durga

I updated the branch to the latest MAUS release and I am seeing the same error in the unit tests as above. To reiterate the returned message

```
Test MapCppTOFDigits process method ... terminate called after throwing an instance of 'MAUS::Exception'
what(): Attempting to find Json property tof_key but not an object at JsonWrapper::GetPropertyStrict
(blah)/test/unit_test.bash: line 32: 1354 Aborted python ${MAUS_THIRD_PARTY}/third_party/install/bin/nosetests -v ${MAUS_ROOT_DIR}/build
```

Still not sure what is going on here. Durga has volunteered to help. The current version of the code is at lp:~ryan-bayes/maus/devel.

#### #27 - 02 May 2014 00:19 - Rajaram, Durga

- File tests.log.gz added

After branching off of lp:~ryan-bayes/maus/devel:

Fixed datastructure issues [ it was causing MCBranhTest to fail, and what was written couldn't be read back because there was no valid processor ]  
Fixed style issues in MapCppMCTriggerDigi.hh[cc], and python style in the tests

Re-ran tests

```
tests/run_tests.bash
```

- tests/integration/test\_simulation/test\_mc\_trigger/test\_mc\_trigger.py fails

```
=====
FAIL: Check that we can trigger off tofl
-----
Traceback (most recent call last):
  File "/home/durga/1182/tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 112,
in test_tofl_trigger_no_pileup
    self._check_data_structure(spill, trigs_expected)
  File "/home/durga/1182/tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 93, i
n _check_data_structure
    trigs_expected[i][1])
AssertionError: 0L != 4
```

```
=====
FAIL: Check that we can trigger off tofl
-----
Traceback (most recent call last):
  File "/home/durga/1182/tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 124,
in test_tofl_trigger_pileup
    self._check_data_structure(spill, trigs_expected)
  File "/home/durga/1182/tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 81, i
n _check_data_structure
    self.assertEqual(spill.GetMCTriggers().size(), len(trigs_expected))
AssertionError: 1L != 2
```

- tests/integration/test\_simulation/test\_tof/test\_tof.py is also failing because the tof histograms are empty. It must be related to the changes MCTriggerDigi is making to tof\_digit [ in MapCppMCTriggerDigi::correctDigitTimes]. Maybe it'll go away after you fix test\_mc\_trigger, we'll see..

log of run\_tests attached

Pushed to lp:~durga/maus/1182

#### #28 - 02 May 2014 15:53 - Rajaram, Durga

- Assignee changed from Rajaram, Durga to Bayes, Ryan

#### #29 - 09 May 2014 15:55 - Bayes, Ryan

The failures in the tests shown by Durga were the result of a problem with the type given by "spill.GetMCTriggers().size()", fixed by explicitly casting the result of the size method as integers. However, once this correction was made further bugs were revealed as the results of the tests were not behaving as expected. Largely these were in the section of the code dealing with particle pileup and incrementing between pileup particles.

The more significant problem may be related to a recent merge. An important function of the MC trigger code is to update the reconstructed branch to define events based on triggers rather than on particle events. A part of this function is resizing the recon\_events data structure to remove elements that are removed due to pileup or lack of a trigger. This rescaling no longer works; specifically, if I start with 5 events --- three of which trigger --- I still have 5 events after running the trigger, when I should have 3 events in the recon\_events data structure.

The json format is used:

```
_root["recon_events"].resize( _root["mctrigger_event"].size() );
```

Is there a better way to handle this.

### #30 - 17 June 2014 01:11 - Rajaram, Durga

The resizing seems to work,

For e.g.

```
std::cout << ">> revt size before: " << _root["recon_events"].size() << std::endl;
_root["recon_events"].resize(_root["mctrigger_event"].size());
std::cout << ">> revt size after: " << _root["recon_events"].size() << std::endl;
```

outputs (when running python src/map/MapCppMCTriggerDigi/test\_map\_cpp\_mctrigger\_digi.py)

```
>> revt size before: 28
>> revt size after: 3
```

The test expects the size to be 2, however, and is failing. I haven't looked into the contents of the spill if indeed it should be 2 instead of 3

### #31 - 17 June 2014 23:08 - Bayes, Ryan

I have reproduced what you have done here, and I think that 3 is the correct size ... I have since found and corrected a bug that lead the code to produce 2 events.

The problem is when I run "python tests/integration/test\_simulation/test\_mc\_trigger/test\_mc\_trigger.py". It tests the mc trigger in a full simulation of a small number of particles, as opposed to test\_map\_cpp\_mctrigger\_digi.py which only tests the mc trigger using pre-defined data. Furthermore it is meant as a test of the data structure, which is what I guess the source of the problem. If I run Durga's test of adding the print statements, I get the expected results -- that is the number of reconstructed events match the number of trigger events, and the number of reconstructed events drops from 5 to 3. These numbers are taken from the json document, however, and it does not seem to be communicated to the root based data structure. For reference I have attached the test output in full below.

```
[rbayes@ppepc141 devel]$ python tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py
FFF
=====
FAIL: test_tof0tof1_trigger_no_pileup (__main__.PhysicsModelTest)
-----
Traceback (most recent call last):
  File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 141, in test_tof0tof1_trig
ger_no_pileup
    self._check_data_structure(spill, trigs_expected)
  File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 83, in _check_data_structu
re
    self.assertEqual(int(spill.GetReconEvents().size()), len(trigs_expected))
AssertionError: 5 != 2
=====
FAIL: test_tof1_trigger_no_pileup (__main__.PhysicsModelTest)
-----
Traceback (most recent call last):
  File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 115, in test_tof1_trigger_
no_pileup
    self._check_data_structure(spill, trigs_expected)
  File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 83, in _check_data_structu
re
    self.assertEqual(int(spill.GetReconEvents().size()), len(trigs_expected))
AssertionError: 5 != 3
=====
FAIL: test_tof1_trigger_pileup (__main__.PhysicsModelTest)
-----
Traceback (most recent call last):
  File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 128, in test_tof1_trigger_
```

```
pileup
    self._check_data_structure(spill, trigs_expected)
File "tests/integration/test_simulation/test_mc_trigger/test_mc_trigger.py", line 83, in _check_data_structu
re
    self.assertEqual(int(spill.GetReconEvents().size()), len(trigs_expected))
AssertionError: 5 != 2
```

-----  
Ran 3 tests in 67.380s

I guess that the solution might be to move from the json formalism to the root formalism and hope that the resizing mechanism there will work better.

### #32 - 18 June 2014 14:44 - Rajaram, Durga

Hm yes I see, but when I run tests/integration/test\_simulation/test\_mc\_trigger/test\_mc\_trigger.py, the print out gives me a "size after = 0" suggesting it didn't trigger at all. Going away from JSON to Cpp may make things cleaner. The new API from Chris allows taking MAUS::Data  
Do you want me to do the modifications?

### #33 - 18 June 2014 15:52 - Bayes, Ryan

I think that the reason you are getting no triggers with tests/integration/test\_simulation/test\_mc\_trigger/test\_mc\_trigger.py is because a recent merge wiped out a change that I had made to the TOF trigger digitization which wrote the tof digits to the MC branch instead of the reconstructed branch. The trigger digitization, in the absence of TOF digits in the MC branch, will produce zero triggers, and forced me to spend the better part of an afternoon in an effort to identify the cause. I will make the changes related to the data structure ASAP to see if this fixes the problem.

### Files

---

MCTriggerStructure.pdf	24.2 KB	22 February 2013	Bayes, Ryan
cpp_coverage.tar.gz	6.2 MB	28 August 2013	Rogers, Chris
tests.log.gz	39.6 KB	01 May 2014	Rajaram, Durga