Monitoring update

Tom Coates, Remi Ete,
Antoine Pingault, Fabrizio Salvatore

University of Sussex

DREAM Testbeam Meeting

2018-06-28

SiPM data

- Started using the provided `ntuplizeMADA.cpp` file used to convert from txt to ROOT – but found it's actually simpler to implement file reading in DQM4HEP
- Implemented `SiPMFileReader.cc`, a DQM4HEP plugin that reads the txt file line-by-line and creates a GenericEvent for each line
- Also separates the XML header of the file from the data – DQM4HEP has a fully-featured XML library, so all information in the header can be added into the event, or into the run information (not yet implemented!)
- Code is very simple and short, hopefully clear to understand and adapt

SiPM data

```
StatusCode SiPMFileReader::open(const std::string &fname) {

  std::FILE *p_dataFile = std::fopen(fname.c_str(), "rb");
  bool isFileOpenable = p_dataFile;

  if(!isFileOpenable) {
    dqm_error("The file at {0} could not be opened.", fname);
    throw StatusCodeException(STATUS_CODE_FAILURE);
  }

  std::string rawData;
  std::fseek(p_dataFile, 0, SEEK_END);
  rawData.resize(std::ftell(p_dataFile));
  std::rewind(p_dataFile);
  std::fread(&rawData[0], 1, rawData.size(), p_dataFile);
  std::fclose(p_dataFile);

  std::string headerTagOpen  = "<ACQUISITION_INFO>";
  std::string headerTagClose = "</ACQUISITION_INFO>";
  std::string dataTagStart   = "</START_NOTE>";

  int headerStartPos = rawData.find(headerTagOpen);
  int headerEndPos   = rawData.find(headerTagOpen) + headerTagOpen.size() - headerStartPos;
  int dataStartPos   = rawData.find(dataTagStart)  + dataTagStart.size();

  headerStream.str(rawData.substr(headerStartPos, headerEndPos));
  dataStream.str(rawData.substr(dataStartPos));

  return STATUS_CODE_SUCCESS;
}
```

```
StatusCode SiPMFileReader::readNextEvent() {
  EventPtr event = GenericEvent::make_shared();
  GenericEvent *generic = event->getEvent<GenericEvent>();

  std::string eventDelimiter = ";";
  std::string currentEventString;

  std::getline(dataStream, currentEventString);

  if (currentEventString.size() <= 1) {
    dqm_warning("Event is blank");
    return STATUS_CODE_SUCCESS;
  }

  dqm4hep::core::tokenize(currentEventString, eventContainer, eventDelimiter);

  if (eventContainer.size() != 64) {
    dqm_error("Event has wrong number of members");
    throw StatusCodeException(STATUS_CODE_FAILURE);
  }

  std::vector<float> ev_eventNum = {eventContainer[0]};
  generic->setValues("Event", ev_eventNum);
  std::vector<float> ev_time = {eventContainer[0]};
  generic->setValues("Time", ev_time);
  eventContainer.erase(eventContainer.begin(),eventContainer.begin()+1);
  generic->setValues("Channels", eventContainer);

  onEventRead().emit(event);
  return STATUS_CODE_SUCCESS;
}
```

University of Sussex

DESY

UNIVERSITEIT GENT

IPNL

AIDA 2020

SiPM data

- Next step is to implement analysis modules, to create plots and graphs (monitor elements):
  - Hitmaps (in progress)
  - Whole-run, per-channel spectra
  - Others?
- Showing results from the analysis module will be a little tricky at the moment – the monitor GUI is still under very active development by Remi, but hopefully able to show non-dummy plots soon!
- Implementing analysis modules is quick, especially since we have existing samples of similar modules from AHCAL testbeams

Summary

- Good progress on SiPM data – decoding and emitting within DQM4HEP
- First analysis modules nearly finished!
- Calorimeter data not started, but should be just as quick!
- All code under development found at https://github.com/tcoates3/dqm4hep-dream

Thank you

University of Sussex