Introduction to the ILD Software Framework
Mokka and Marlin

January 19th & 20th, 2009

Things to be Covered

- Today
  - Running Mokka
  - Simulating Events with Mokka
  - Changing Detector Geometry
- Tomorrow
  - Reconstruction & Analysis with Marlin
  - Running Marlin
  - Writing your own Processor

Mokka in a Nutshell

- Geant4 Based
- Subdetectors (Vertex, Calos, TPC, tube, mask, magnetic field) described by C++ Drivers
- Parameters stored in a Database
- Output from Mokka
  - LCIO file containing MCParticles and Hits in the Subdetectors
  - XML File (Gear) containing geometry used for simulation

Running Mokka

- Mokka can be told what geometry to build using a steering file
- Which Database to use, user, password, Detector model, changes in Geometry, physics list, name of output files, …
- Then either use the Geant4-Shell or a Geant4 Macro

- For this tutorial get this file and extract:

wget http://www.cern.ch/lcd-data/software/MokkaTutorial.tgz

Sample Steering file

MYSQL SERVER
/Mokka/init/dbHost pccds03.cern.ch
/Mokka/init/user consult
/Mokka/init/dbPasswd consult
DETECTOR MODEL
/Mokka/init/detectorModel CLIC01_ILD
/Mokka/init/physicsListName QGSP_BERT
Verbosity / Batch Mode
/Mokka/init/printLevel 0
/Mokka/init/BatchMode false
/Mokka/init/initialMacroFile opengraphics.mac
Changing the Geometry
#/Mokka/init/EditGeometry/rmSubDetector all
#/Mokka/init/EditGeometry/addSubDetector SHcalSc02 300
#/Mokka/init/globalModelParameter Hcal_nlayers 30
OUTPUT OF SLCIO/ASCII
/Mokka/init/lcioFilename out.slcio
/Mokka/init/lcioWriteMode WRITE_NEW

Initial Output from Mokka

- Let's have a look at the output from Mokka
- …
- Information on the Detector Model
- Information for all the subdetectors used
- Information on the Physics list used

Database

MYSQL SERVER

/Mokka/init/dbHost pccds03.cern.ch

/Mokka/init/user consult

/Mokka/init/dbPasswd consult

Set the database, the one given here is my PC

Default is polui01.in2p3.fr

User and Password is also needed, normally the values consult//consult are used

Detector Model

DETECTOR MODEL
/Mokka/init/detectorModel CLIC01_ILD
/Mokka/init/physicsListName QGSP_BERT
/Mokka/init/rangeCut 0.005 mm

Tell Mokka which Detector Model to use
Physics list (default is LCPhys)
Set the Geant4 RangeCut (default is 0.005 mm)

/Mokka/init/printLevel 0

Define the amount of output

/Mokka/init/BatchMode false

If true, Mokka exits after the macro file has been run

/Mokka/init/initialMacroFile opengraphics.mac

Initial Macro file to be run, once the geometry is build

/Mokka/init/lcioFilename out.slcio

Name of the LCIO output file

/Mokka/init/lcioWriteMode WRITE_NEW

If the output file exists it is replaced by the new file

Changing the Geometry

Only possible within the steering file

/Mokka/init/EditGeometry/rmSubDetector all

Remove all subdetectors, or give a specific one

(There is no warning if you misspell the name)

/Mokka/init/EditGeometry/addSubDetector SHcalSc02 300

Add a subdetector, second input is the build order

(There is no warning if you have twice the same subdetector type)

/Mokka/init/globalModelParameter Hcal_nlayers 30

Change one of the many Parameters in Mokka

Adrian Vogels
Detector Model Database Browser

- http://www-flc.desy.de/ldcoptimization/tools/mokkamodels.php
- Select the Detector you are working on
  - Sorry the CLIC01_ILD Model is not yet there, ILD_00(fw) or CILD_00(fw) are very close
    - fw = forward: More details in forward region, for ILC this means also Fieldmap + AntiDID
- All the subdetectors and their parameters are listed

Things to know

- Normally Mokka scales the detector correctly depending on the things you change
- Subdetectors tell other subdetectors how large they are
- The order the subdetectors are build matters!
- If you remove a subdetector the rest might be different!

Other Mokka Commands

- /Mokka/init/TPCCut 10 MeV #Affects what particles cause hits in trackers
- If I understand correctly: only particles generated inside the tracking region are saved in the MCParticle collection
- /Mokka/init/globalModelParameter tracker_region_zmax 2400
- /Mokka/init/globalModelParameter tracker_region_rmax 2400
- I forgot what this was used for
- /Mokka/init/globalModelParameter calorimeter_region_zmax 2400
- /Mokka/init/globalModelParameter calorimeter_region_rmax 2400

Geant4 Visualization Commands

```
###opengraphics.mac
/vis/open OGLSX 600x600+600+400
/vis/scene/create
/vis/drawVolume
/vis/scene/add/trajectories
/vis/scene/add/hits
/vis/viewer/flush
/control/execute othermacro.mac
```

More Commands

- /vis/viewer/set/viewpointVector 0 1 0
- /vis/viewer/set/viewpointThetaphi 45 45
- /vis/viewer/set/upVector 1 0 0
- /vis/viewer/set/sectionPlane on 0 0 0 m 0 1 0
- /vis/viewer/addCutawayPlane 0 0 0 m 0 -1 0
- /vis/viewer/zoom(To) 10
- /vis/viewer/pan(To)3 0 m
- help

Links

- A. Vogels MAQs: http://www.desy.de/~vogel/maq.htm
- Generator Files for CLIC Benchmark Processes: http://lcd.web.cern.ch/LCD/Documents/GeneratorFiles.html
- Linear Collider Forum: http://forum.linearcollider.org/
- ILCSoft: http://ilcsoft.desy.de/portal
- Mokka: http://polzope.in2p3.fr:8081/MOKKA/
- Mailing lists: http://www.ilcild.org/mailing-lists
- Also see the ReleaseNotes for information regarding drivers

Mokka on the Cern Batchfarm

- For historical reasons I use a tarball of the executable and libraries and ship it to the batch node
- See the MokkaBatch script
  - Sets up environment, creates steering and macro file and writes everything to castor at the end

Sample macro File

- Using the Particle Gun

  /gun/particle mu-
  /gun/energy 300 MeV
  /gun/momentumSmearing 50 MeV
  /gun/direction 0 0 1
  /run/beamOn 1000

- Using Events generated elsewhere (stdhep or HEPEvt input File, guineapig.pairs will also be accepted, but this is very GP version dependent)

  /generator/generator mmnn02_3TeV.stdhep
  /run/beamOn 10

Marlin

- Modular Analysis and Reconstruction for the LINear Collider
- Modular = Plug and Play Processors
  - Digitization, Track Reconstruction, Clustering, Particle Flow, Particle ID, …
- Tell Marlin which lcio file(s) to open (sequentially) and what Processors to run with given parameters
- XML Steering file as input
  - Marlin –c steering.xml Checks steering file
  - Marlin steering.xml Runs steering file

Example Steering File

```
<marlin xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:noNamespaceSchemaLocation="http://ilcsoft.desy.de/marlin/marlin.xsd">
        <execute>
                <processor name="MyMarlinExample"/>
                <processor name="MyMarlinExample2"/>
        </execute>
         <global>
                <parameter name="LCIOInputFiles"> muons.slcio </parameter>
                <parameter name="MaxRecordNumber" value="" />
                <parameter name="SkipNEvents" value="0" />
                <parameter name="SupressCheck" value="false" />
                <parameter name="GearXMLFile"> GearCLIC01_ILD.xml </parameter>
                <parameter name="Verbosity“ > WARNING </parameter>
        </global>
         <processor name="MyMarlinExample" type="MarlinExample">
                <parameter name="CollectionName" type="string" lcioInType="MCParticle">
                MCParticle </parameter>
        </processor>
         <processor name="MyMarlinExample2" type="MarlinExample">
                <parameter name="CollectionName" type="string" lcioInType="MCParticle">
                MCParticle2 </parameter>
        </processor>
</marlin>
```

Marlin Processor in a Nutshell

- Processor::init() Run once in the beginning
  - Initialize Histograms, trees etc.
- Processor::processEvent() Run for each event
  - Fill Histograms, trees etc.
- Processor::end()
  - Save your histograms, trees etc.
- Processor::processRunHeader()
  - Is run whenever a new lcio file is opened (I think)

Create your own Processor

- Use the CopyProcessor.csh script
- Changes the Name of the Processor and also the Name of the Class etc.
- Change the code (See LCIO Reference: http://lcio.desy.de/v01-11/doc/doxygen_api/html/index.html)
- Compile

  cd build

  cmake –C ../BuildSetup.cmake ..

  make install
- If you add external dependencies, add them in the CMakeLists.txt
- Before Running your processor in marlin, add the location of the library to the environment variable MARLIN_DLL

More Things to know

- The lcio commands:
  - dumpevent: detailed output of a single event
    - Note that a maximum of 1000 Elements of a collection are printed
  - anajob: information about a given LCIO file

Day 2

LCIO Collections from Simulation

- MCParticle: As an input from the physics generator (Status 1), or generated during the simulation (Status 0)
- SimTrackerHit, SimCalorimeterHit: Simulated hits written by the Sensitive Detectors
  - (Turn the verbosity of Mokka up to 1 or 2?) and you are informed how many hits there are per collection
  - Use anajob to see all the collections that have hits

ILD Standard Reconstruction

- Get the file from here or from Indico:

  http://www.cern.ch/lcd-data/software/MarlinStdReco.tgz

- Contains Steering file, Gear File and a few Muons simulated in Mokka
- This file is somewhat old, Marco and Jean-Jacques have updated the reconstruction for LCD (higher energies)
- The PandoraPFA will be replaced by a newer version soon

ILD Standard Reconstruction

- Standard Reconstruction using processors in MarlinReco et al.
  - Digitization
  - Track Fitting
  - PandoraPFA
- Output in DST File
  - Reconstructed Particles
    - PFObjects connected to clusters and tracks)
    - Jet Clustering
  - Clusters, Tracks
  - LCRelation: Relation between Reconstructed Particles and MCParticles
- REC File
  - Also Contains all the Simulated and Digitized Hit collections and all MCParticles

CED Viewer

- If you want to visualize the collections
- In the CernVM launch “glced” in a separate shell
- Use the view.xml to draw events
- Keys 1-0 to turn collection types on and off
- Shift-1 turns the vertex detector off
- Left Mouse: Rotate
- Middle Mouse: Pan
- Right Mouse: Zoom