Compact Muon Solenoid

جـــــامعة الفــــيوم
Fayoum University

CMSSW: Event Generator

Presented by

Dr.Mohammed Attia Mahmoud

-FSQ-Gen contact, CMS Experiment

-PhD from Fayoum University, and Antwerp University, Belgium

-Lecturer of Physics, Fayoum University

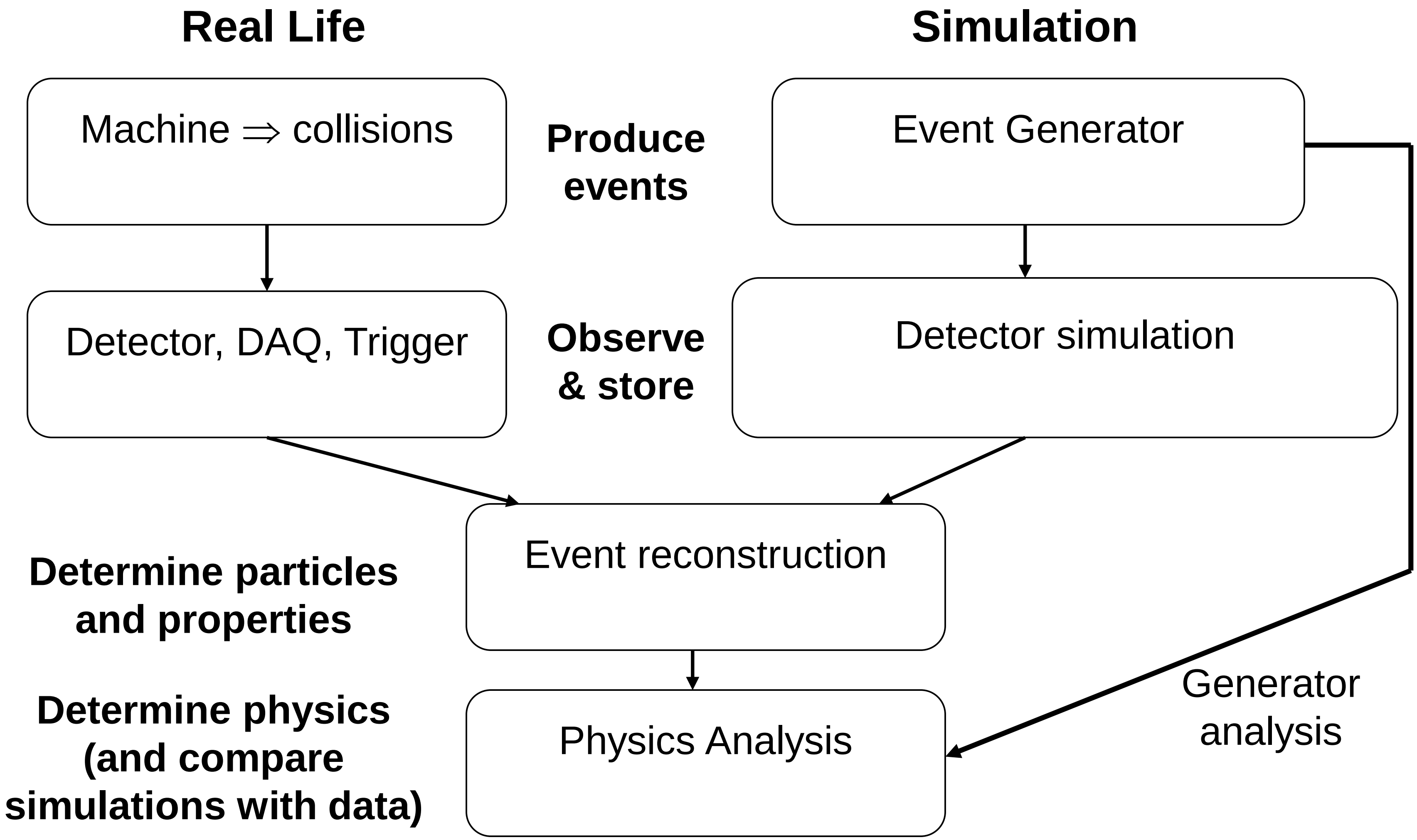
-Researcher at CTP, BUE

BUE
The British University In Egypt
الجامعة البريطانية في مصر

Outlines

- The Simulation Process
- Event generator, Why event generator?
- CMSSW and Event Generators
- CMSSW Intro
- Composing Full Simulation

The Simulation Process



Real Life

Simulation

Machine ⇒ collisions

Produce events

Event Generator

Detector, DAQ, Trigger

Observe & store

Detector simulation

Determine particles and properties

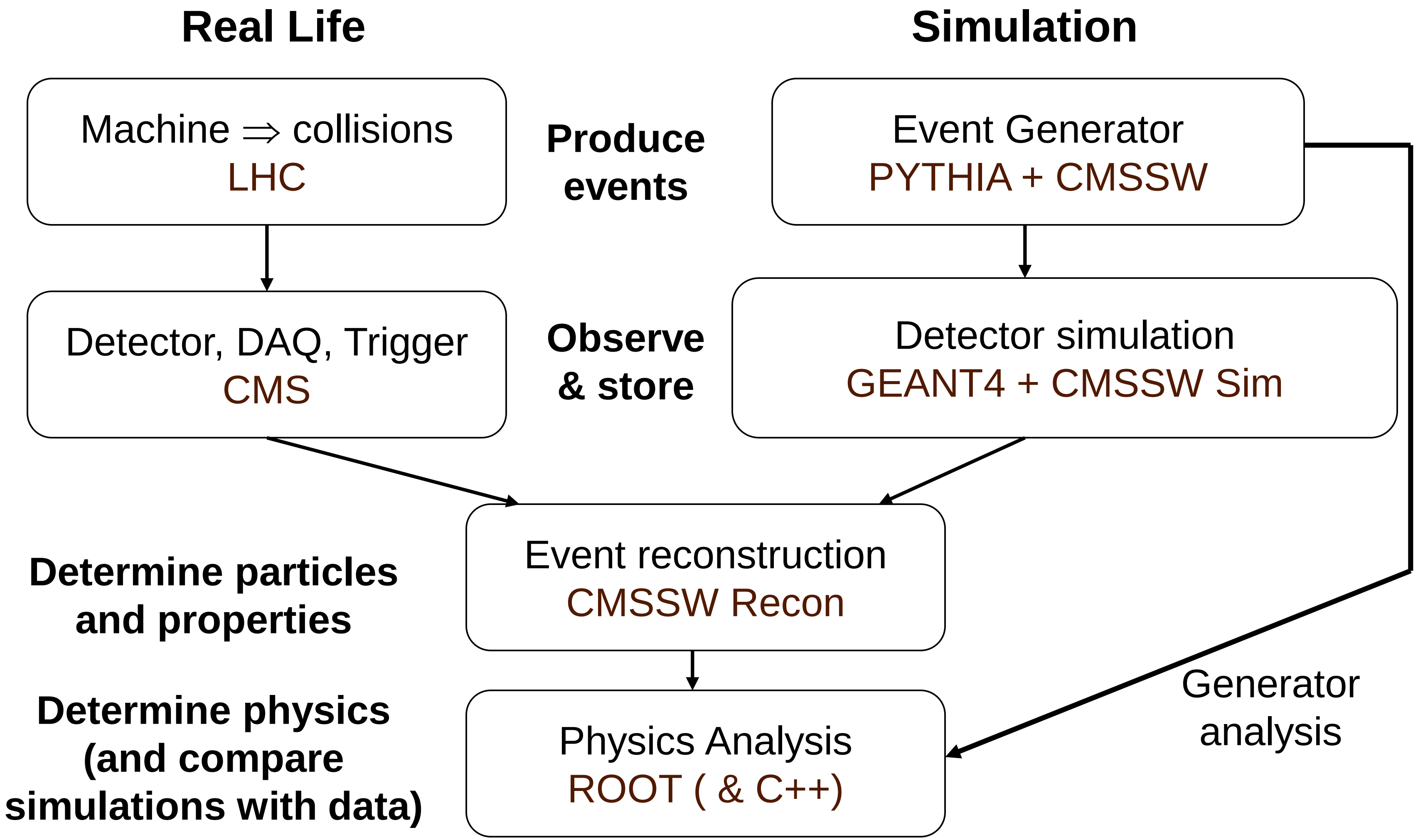
Event reconstruction

Determine physics (and compare simulations with data)

Physics Analysis

Generator analysis

The Simulation Process



Real Life

Simulation

Machine ⇒ collisions
LHC

Produce events

Event Generator
PYTHIA + CMSSW

Detector, DAQ, Trigger
CMS

Observe & store

Detector simulation
GEANT4 + CMSSW Sim

Determine particles and properties

Event reconstruction
CMSSW Recon

Determine physics (and compare simulations with data)

Physics Analysis
ROOT (& C++)

Generator analysis

Event Generator

are software libraries that generate simulated high-energy particle physics events. They randomly generate events as those produced in particle accelerators , collider experiments or during the initial phases of the Universe creation. Events come in different types called *processes*.

Why event generator?

- To give physicists a feeling for the kind of events one may expect/hope to find, and at what rates.
- As a help in the planning of a new detector, so that detector performance is optimized, within other constraints, for the study of interesting physics scenarios.
- As a tool for devising the analysis strategies that should be used on real data, so that signal-to-background conditions are optimized.
- As a method for estimating detector acceptance corrections that have to be applied to raw data, in order to extract the 'true' physics signal.
- As a convenient framework within which to interpret the observed phenomena in terms of a more fundamental underlying theory (usually the Standard Model).

Some Event Generators

- **PYTHIA:** high-energy pp and pbarp collisions. Also e+e- and mu+mu- annihilation processes may be simulated.
- **Herwig++:** Herwig++ is a general-purpose event generator for the simulation of high-energy lepton-lepton, lepton-hadron and hadron-hadron collisions with special emphasis on the accurate simulation ofQCD radiation.
- **ISAJET:** is a Monte Carlo program which simulates p p, pbar p, and e+ e- interactions at high energies. It is based on perturbative QCD plus phenomenological models for parton and beam jet fragmentation.
- **HYDJET++:** is the event generator to simulate relativistic heavy ion AA collisions as a superposition of the soft, hydro-type state and the hard state resulting from multi-parton fragmentation.
- **PYQUEN:** is event generator for simulation of rescattering, radiative and collisional energy loss of hard partons in expanding quark-gluon plasma created in ultrarelativistic heavy ion AA collisions (implemented as modification of standard pythia6.4xx jet event).
- **HYDRO:** is fast event generator for simulation of ``thermal'' hadronic spectra (including radial and elliptic hydro-flow effects) in central and semi-central heavy ion AA collisions at LHC energies

CMSSW and Event Generators

An impressive collection of physics event generators are interfaced to CMSSW.

- They include several **general-purpose generators, such as Pythia8 and Herwig++ and other specific** to some processes like EvtGen and Tauola. The objective of general-purpose event generators is to provide as accurate as possible a description of what happens end-to-end in a hadron collision. They contain theory models for a number of physics aspects, such as hard and soft interactions, parton distributions, initial and final state parton showers, multiple interactions, fragmentation and decay.

- **Matrix Element (ME) calculators, such as Powheg,** MadGraph5_aMCatNLO, Algpen etc. They deliver an event at the parton level, and one or another multi-purpose generator can further be used to develop a fully hadronized event.

- In addition, we have several specific generators for diffractive physics, cosmic muon generators, heavy ions and so on.

- For the software testing purposes and acceptance studies, there is also a collection of simpler tools, so called particles guns, that allow to generate one or many individual particles of the user-specified kinematics.

Let's START

CMSSW Intro

- Login to your lxplus:

```
ssh –Y  username@lxplus.cern.ch
```

- Listing the available CMSSW:

```
scram list
```

- Choosing CMSSW By using this command:

```
cmsrel    CMSSW_X_Y_Z
```

Inter into CMSSW_X_Y_Z/src by using :

```
cd SSW_X_Y_Z/src
```

- Apply CMS environment :

```
cmsenv
```

- Running generation and simulation in CMSSW

```
cmsRun <My configuration file>
```

Composing Full Simulation

Standard CMSSW utility called cmsDriver.py.

briefly review the input arguments to cmsDriver.py utility:

1. configuration fragment that determines what physics event generator you wish to use and what topology you intend to generate. In this example we use a pre-fabricated fragment which is originally located in the genproductions area.

2. The **-s** field contains the sequence of event processing steps. The chain starts with the **GEN**(eration), including necessary filters to select events
   SIM(ulation),
   DIGI(tization),
   L1 trigger emulation,
   DIGI2RAW conversion of the simulated (raw data format),
   HLT and **H**(igh)**L**(evel)**T**(triggers) simulation and
   RECO(nstruction).

3. **--conditions** field can be found in the Software Guide on the FrontierConditions. Choice --conditions **auto:mc** chooses the best conditions for a given release.

4. **--datatier =>GEN, GEN-SIM, GEN-SIM-RAW, GEN-SIM-RECO**

5. **--eventcontent** field is described in great details in the SWGuideDataFormatTable.

6. **-n** field you will specify how many events you want to generate, simulate, etc.

7. **--no_exec** argument tells cmsDriver.py to write out the configuration file. If you do not specify this argument, cmsDriver.py will proceed to executing cmsRun.

8. **--python_filename** and **--fileout** define the output configuration file and the output root file after cmsRun, respectively.

Example:

https://github.com/cms-sw/cmssw

❑

```
cmsDriver.py  Configuration/Generator/python/choose_File_cfi.py  -s GEN,SIM  --conditions auto:mc   -n 100 --no_exec  --python_filename  produced_Config_cfg.py --fileout output_root_file.root
```

❑ After run the above command, you will see, the new python file will create.

❑ You can open this new python file by any editor such as: pico, vi, vim, emacs, xemacs, gedit ,…………………………………………

❑ You will see inside the python the following lines:

O First of all, please notice this line:

```
process.source = cms.Source("EmptySource")
```

You will need to use EmptySource if you wish to generate events "from scratch", using one of the multi-purpose event generators.

O Next block in the configuration file that will be of interest to you is the one that starts with

```
process.generator = cms.EDFilter("Pythia8GeneratorFilter",
```

followed by a long string of configuration commands. This is the software module of CMSSW that interfaces Pythia8 multi-purpose event generator.

O Towards the beginning of the configuration file you will see other topics of interest to you:

```
process.load('Configuration.StandardSequences.SimIdeal_cff')
```

for Geant4-based detector simulation

O Towards the end of the configuration file you will notice how these labels are used to include these software in the processing chain:

```
process.simulation_step = cms.Path(process.psim)
```

```
cmsRun configuation_file_cfg.py
```

When the "cmsRun" step has completed, then perform:

```
edmDumpEventContent          Output_file.root
```

>>>>>Explain what will you see???????

Unprocessed information about the events can be found in the GenEventInfoProduct "generator" and the **GenParticle** collection "**genParticles**".

Reconstruction step

```
cmsDriver.py step2 -s DIGI,L1,DIGI2RAW,HLT,RAW2DIGI,RECO --conditions auto:mc --datatier
GEN-SIM-RECO --eventcontent FEVTDEBUG -n -1 --no_exec --filein output_root_file.root --
fileout MinBias_DIGI-RAW.root --no_exec --python_filename MinBias_DIGI-RAW-RECO.py
```

Thanks