# Simulation software: the way forward

Frank Gaede, CERN/DESY

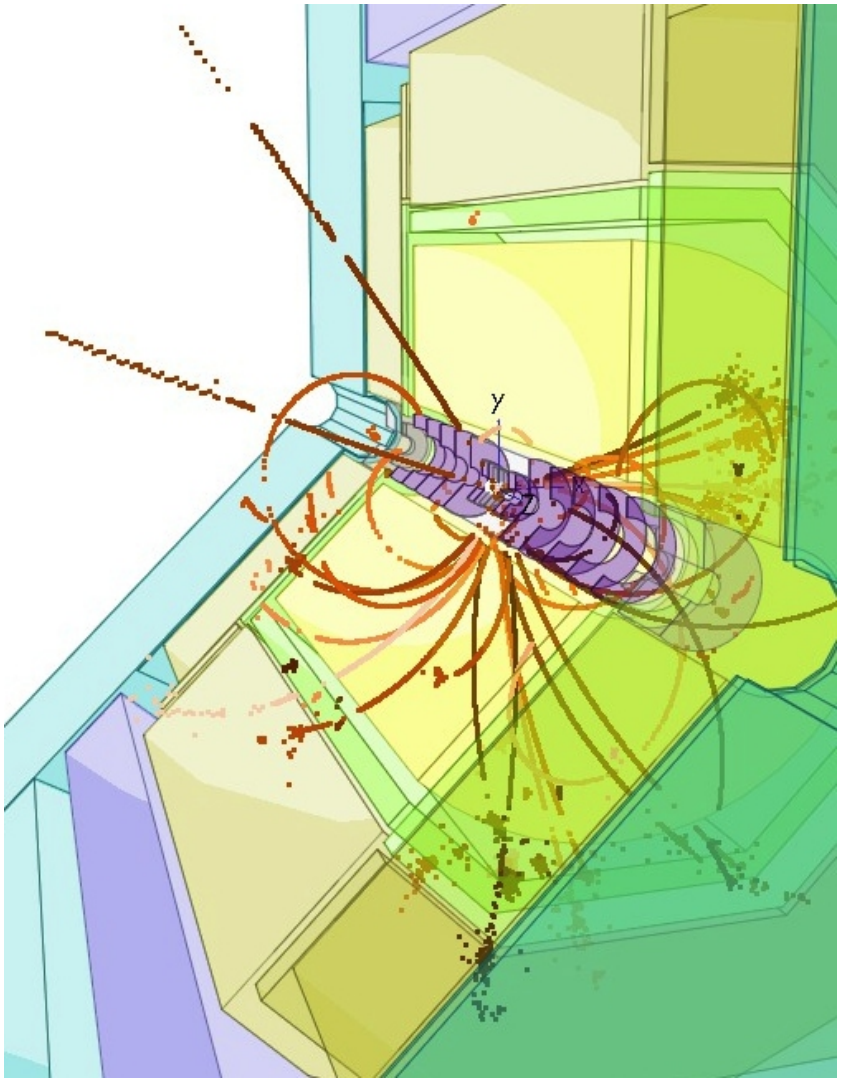
ILD Meeting

LCWS 2014, Belgrade, Serbia

5-10 October 2014

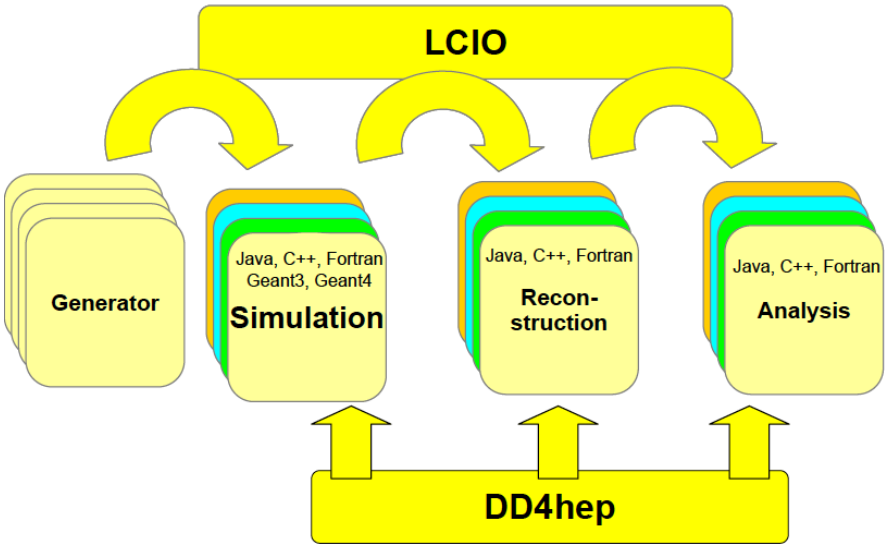
# Outline

- Introduction
- DD4hep, DDG4, DDRec
- Recent developments and status
- ILD_o1_v05 in DDSim
- Towards new ILD simulation models and a timeline
- Summary & Outlook



# Introduction

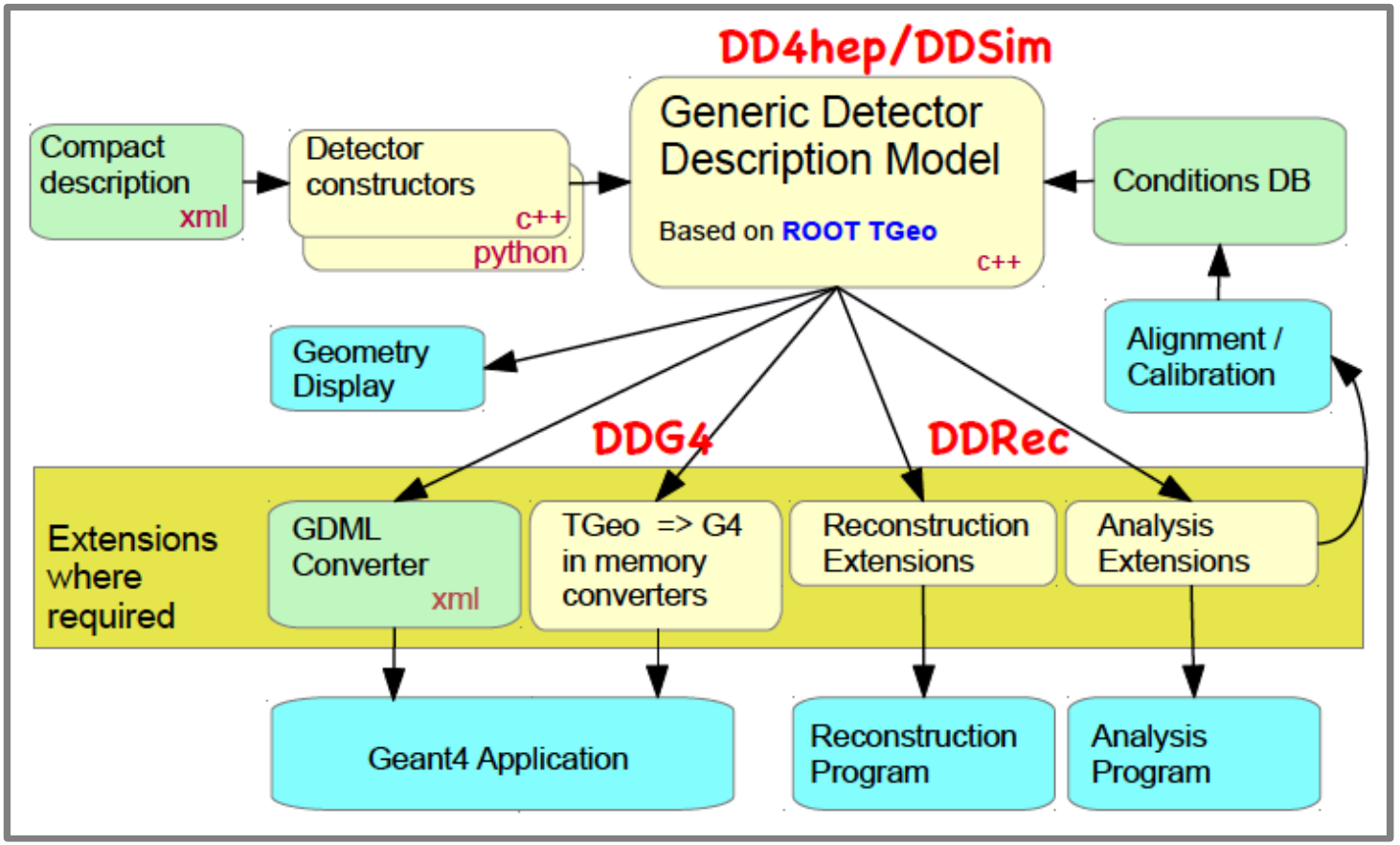
- in Linear Collider Software Meetings 2012/2013 decided to use new detector geometry description **DD4hep** as basis for a new common LC simulation package




- **DD4hep:** common detector geometry description
  - developed in AIDA WP2 (CERN, DESY)
- will be used by CLICdp, FCC and ILD,...

- defining a common geometry API is the second step - after the common EDM (LCIO) - that is needed to have an open and modular software framework

# DD4hep - overview



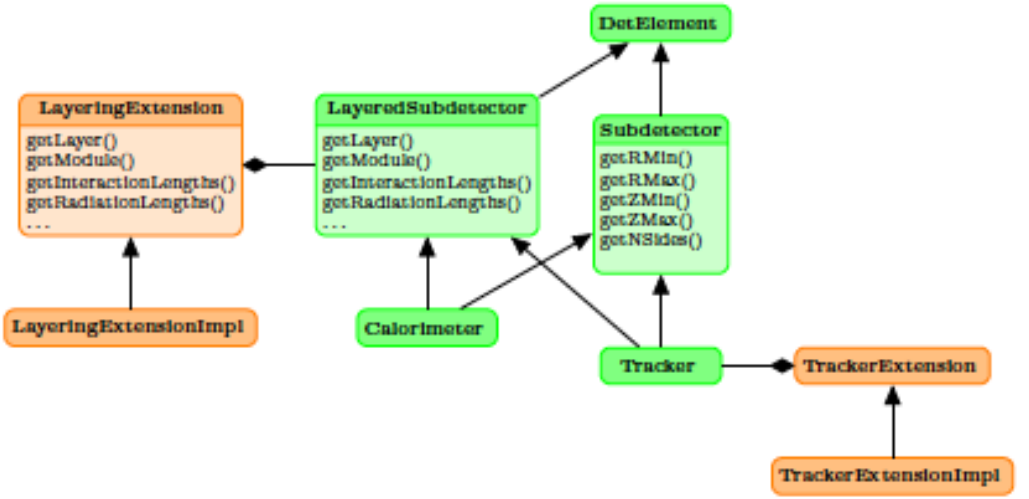

- advantages of DD4hep:
  - consistent description of detector geometry with one unique source
  - possibility to simulate misalignment to study **alignment** strategies for ILD
  - cooperation w/ CLICdp and SiD

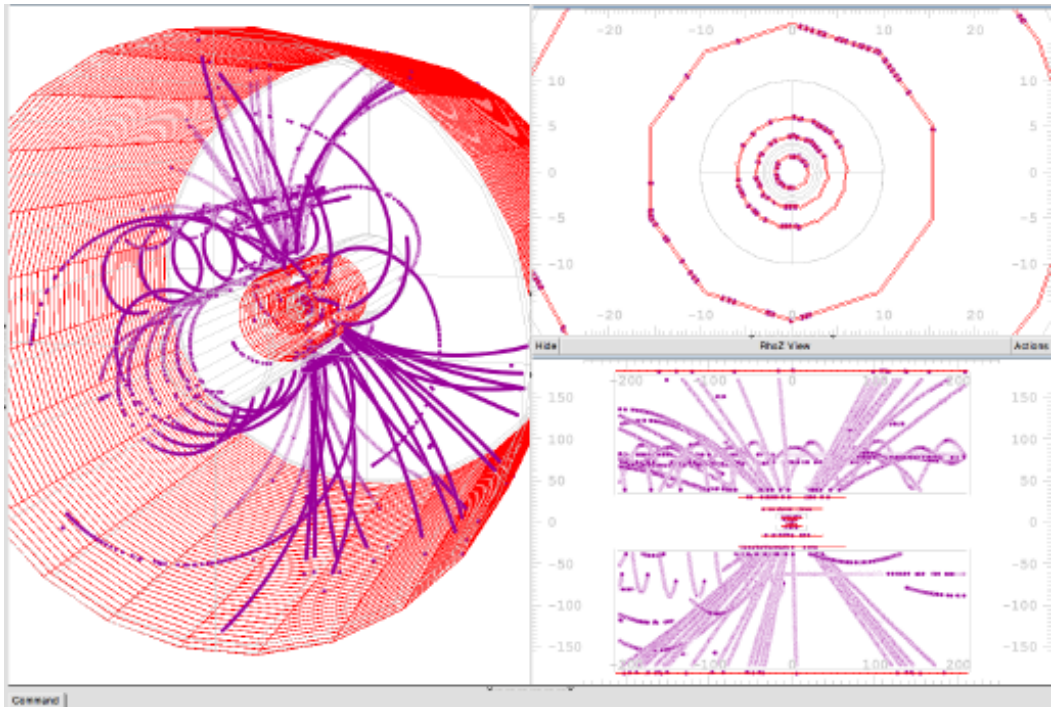
# DDG4 - built in Geant4 gateway

- in memory conversion of TGeo geometry to Geant4 geometry
- modular design using plugin mechanism for
  - sensitive detectors, Geant4 user actions : stepping, tracking,...
  - input (generator files) and output (LCIO,...)
- configure mechanism with xml, python or CINT:
  - physics lists, limits, fields,...
  - define sequences for
    - input, sensitive detectors, user actions, output,...
- features :
  - full flexibility in sensitive detectors
  - can use extension code in simulation and reconstruction
  - supported by CERN for FCC and CLICdp

# DDRec - interface to reconstruction

- extension mechanism is used to define interface for reconstruction
- calorimeters and trackers defined as LayeredSubdetectors
  - use to eventually replace GEAR
  - work in progress ...
- for tracking additional Surfaces provide:
  - u,v,normal and origin
  - inner and outer (averaged) material incl. thickness
- -> planes and cylinders allow for simple navigation in detector geometry for the tracking



# DDRec versus GEAR

- need to be able to run existing reconstruction code MarlinReco, MarlinTrk, PandaraPFA and LCFIPlus with new simulation model
- completely based on GEAR
- -> need for (intermediate) interface from DDRec to GEAR

- current thinking:
- introduce light weight classes that hold relevant data and fill them on detector construction
- write 'quick and dirty' throw away code to generate GEAR objects from DD4hep detector model
- test and run existing reconstruction
- further develop new DDRec classes and eventually
- replace GEAR w/ DDRec in Marlin based reconstruction

# recent developments in DD4hep

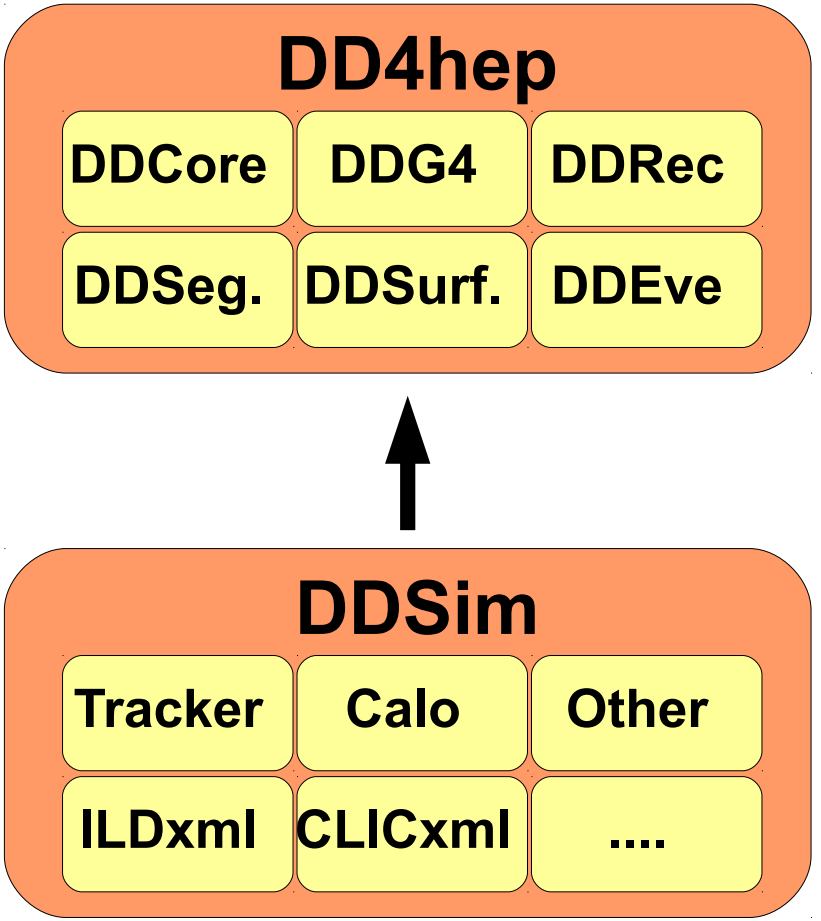
- implemented data classes (EDM) for MC truth and sim tracker/calorimeter hits analogue to LCIO classes:
  - MCParticle, SimTrackerHit, SimCalorimeterHit
- provided input and output modules for LCIO
  - also have stdhep input option, (HepEvt to come)
  - other formats are possibly, e.g. ROOT I/O
- implemented MC-Truth linking algorithm
  - similar algorithm to the one implemented in Mokka
- implemented 'canonical' sensitive detectors for trackers and calorimeters that work with LCIO SimTracker/CalorimeterHits and simple segmentations
- will need a few specialized sensitive detectors (e.g.TPC)

# DDSim simulation package

- created package DDSim as a common LC simulation package for ILD and CLIC (and SiD)
- eventually want to preserve all current Mokka models - started with ILD_o1_v05:
  - extract DB params to xml and
  - line-by-line port of geometry drivers
- will soon add CLIC models from DD4hep/examples
- for details and code:

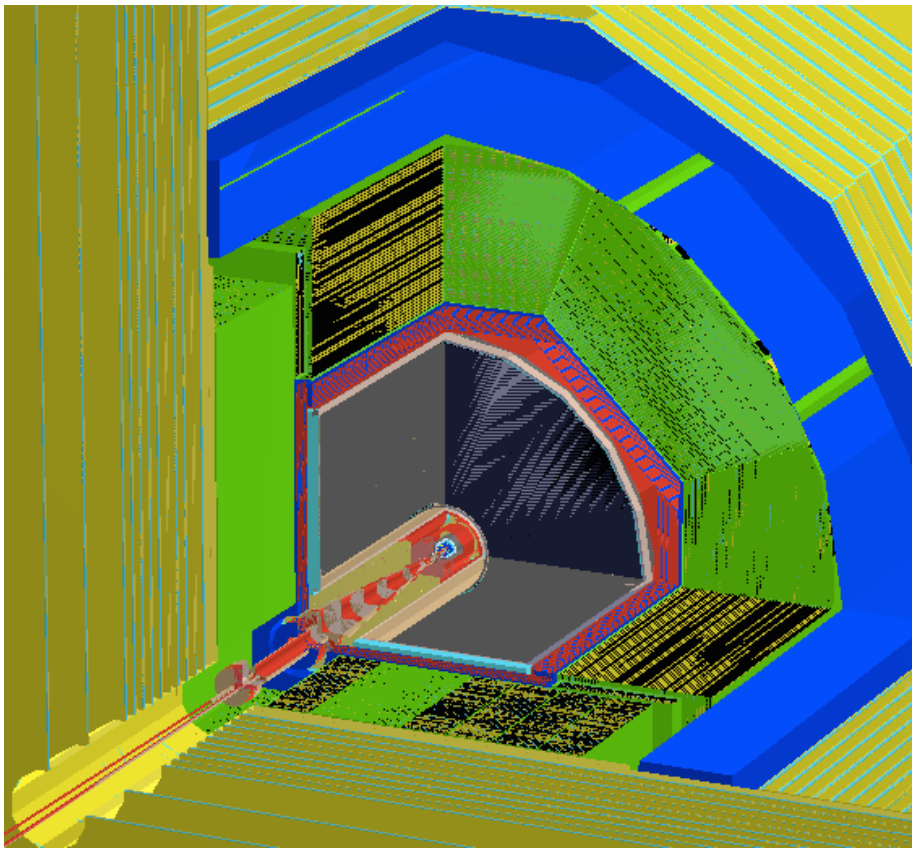
https://svnsrv.desy.de/viewvc/ddsim/DDSim/

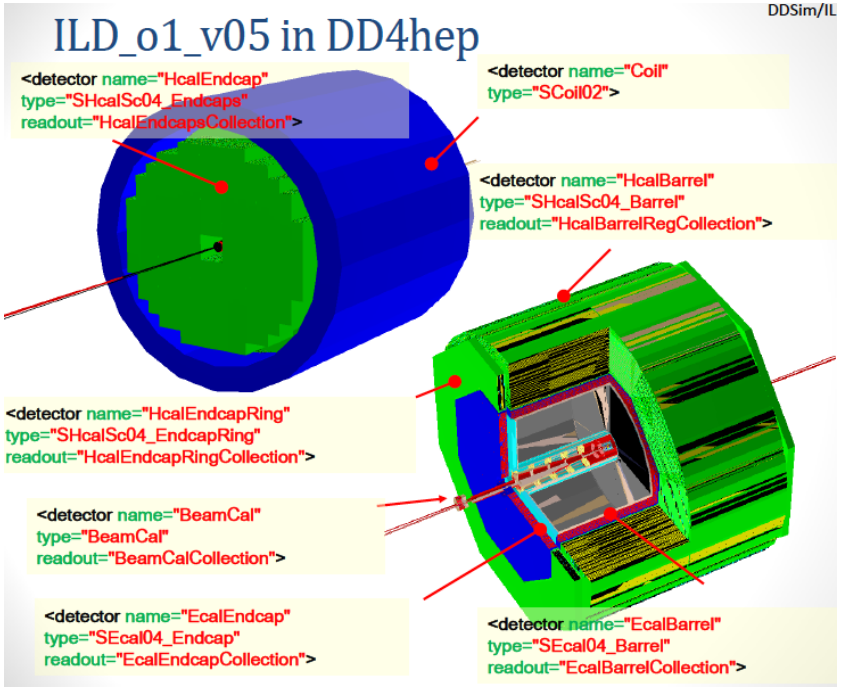
- NB: code and structure still somewhat experimental and targeted at sw-experts

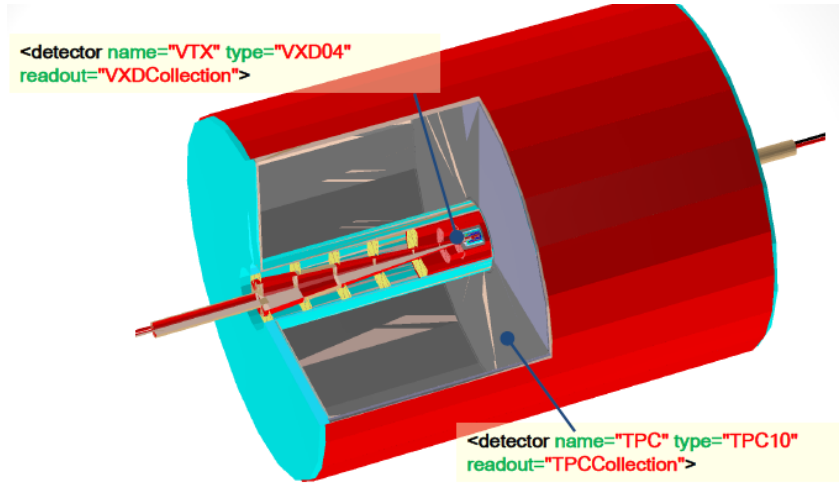


# ILD_o1_v05 in DDSim

- complete Mokka model ILD_o1_v05 ported:
- VXD, FTD, SIT, TPC, SET, beam pipe
- Ecal, Hcal, Yoke (S.Lu)
- Beamcal, Lcal, LHcal (A.Sailer, M.Petric)
- so far using 'canonical' sensitive detectors

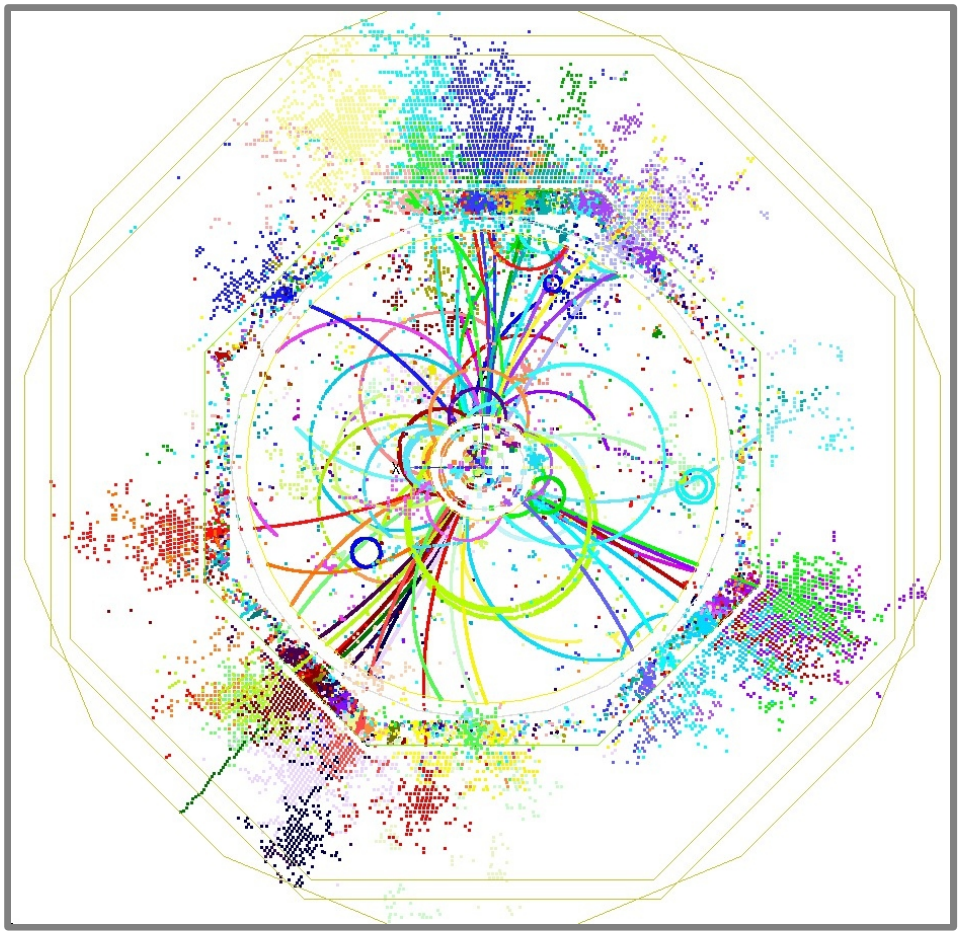




# running the ILD simulation I

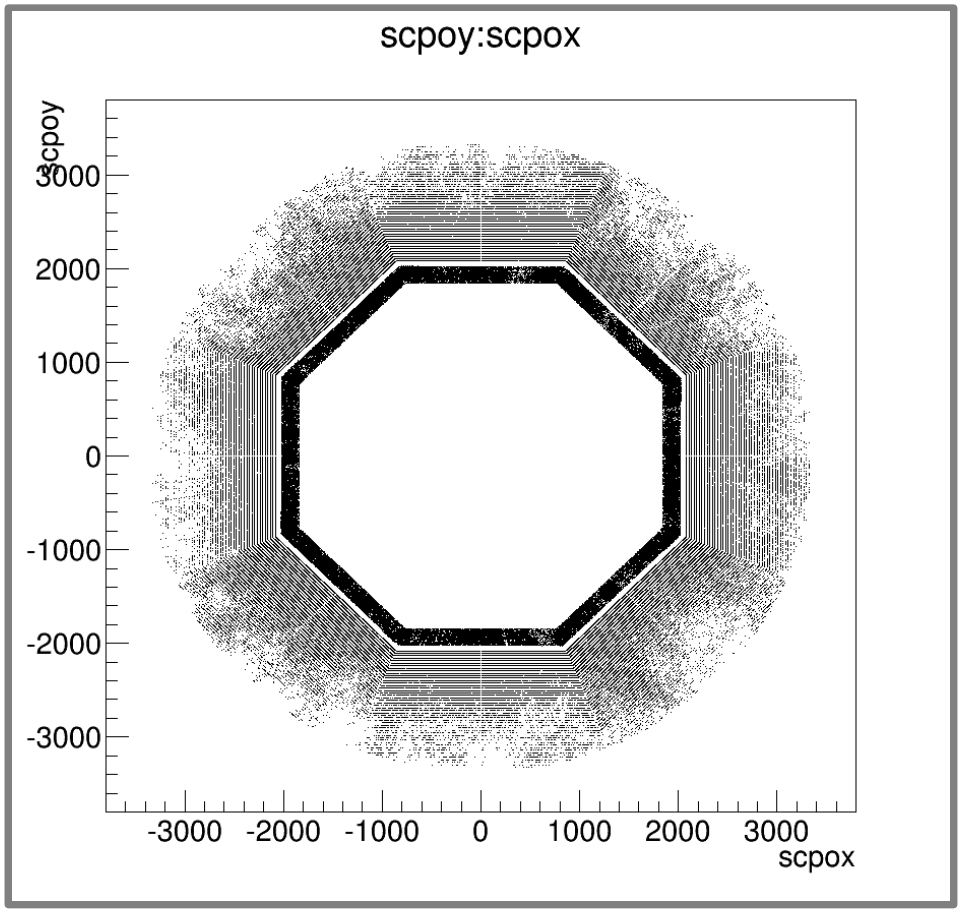




note: hits colored with MC-truth information in Sim-Hits

- we have now for the first time the complete basic functionality implemented in DD4hep to run a simulation:
- 'first' event simulated in DD4hep model of ILD_o1_v05
- ttbar->bbudsc @ 500 GeV
- first look reveals no major problem:
  - need to look in more detail...

# running the ILD simulation II




- first look at hit maps in barrel calorimeter with new ILD_o1_v05
- use standard LCTuple to create hit maps
- work just started

  need to systematically study all sub detectors ...

- need to identify sub detector experts for the simulation models to
- verify the models in close contact with R&D groups

# New ILD simulation models

- in recent ILD meetings it has been discussed and largely agreed to create a small (2-3) number of new ILD simulation models for detector optimization
- ideally we should do this in DD4hep/DDSim !
- Issues involved:
- ILD_o1_v05 model ported from Mokka also has ported all issues from the existing models:
  - large (excessive?) number of parameters
  - run time dependence of parameters
  - partly undefined scaling behavior
  - mostly undefined enclosing volumes (real-estate/place holder)
- these issues should be addressed by another iteration on the sub detector constructors in DDSim
- -> question of timeline and manpower

# Towards a timeline for software I

- ingredients and missing items needed for defining a timeline for ILD software development:
  - need first functional version of ILD_o1_v05 in DD4hep/DDSim ~done
  - need functional interface to existing reconstruction (GEAR/DDRec)
  - need testing and validation
  - define the ILD optimization models - how many ( 2-3 ) ?
    - reference detector + smaller detectors ...
  - implement these models
  - need testing and validation
  - define the physics benchmarks/data samples that need to be processed
    - 250 GeV, 350 GeV, 500 GeV full SM ?
  - finalize the Grid production infrastructure w/ ILCDirac ~done
  - adapt reconstruction to new models
  - need testing and validation
  - estimate the CPU (and storage) needs
  - the actual Monte Carlo mass production
  - ...

# Towards a timeline for software II

- a first very rough estimate of the effort involved:

| item: | estimated effort* | comment |
|---|---|---|
| first version of ILD_o1_v05 in DD4hep | 1 pm | **~done** |
| interface to reconstruction | 1 pm | |
| testing and validation | 2 pm | |
| define ILD optimization models | 1 pm | start at ILD meeting |
| implement these models | 3 pm | # models ? |
| testing and validation | 3 pm | # models |
| define physics benchmarks | 1 pm | |
| Grid production infrastructure | 1 pm | **~done** |
| Grid Monte Carlo simulation | 1 pm (3 months) | # channels/processes, # CPUs |
| adapt reconstruction (incl. testing) | 2 pm | |
| Grid reconstruction | 1 pm (1 month) | |
| Total | 17 pm | |

*pm: full time person month

- calendar time depends on number of (experienced) people
- Not included here: producing generator files and analyses !

# alternative scenarios

- the previous slides assumed a 'full program' for detector optimization with a large scale Monte Carlo production and a reasonably large set of physics benchmarks (scenario 3)

- could speed-up the process by:
- not producing large SM Monte Carlo sets but study detector models with the classical detector performance parameters:
  - tracking efficiency, momentum and impact parameter resolutions, jet energy resolutions, flavor tag performance,...

  (scenario 2)

- using Mokka to define smaller ILD models for initial optimization studies (scenario 1)

# scenario 1

- use Mokka to define smaller ILD models for initial optimization studies
- pros:
  - could start very soon
  - some smaller models already exist and have already been used for optimization studies (tracking and JER)
- cons:
  - existing models have not yet been fully tested and validated
  - not tagged in central DB (needed for reproducibility)
  - would require some effort from software experts that is not available for finalizing new simulation software
  - results in duplication of effort, as eventually will have to redo models in new framework
  - models might not be flexible enough to react to requests from the machine ( smaller L* , reduced crossing angle (?),....)

# scenario 2

- not producing large SM Monte Carlo sets but study new DD4hep detector models with the classical detector performance parameters
- pros:
  - could have first results a few months earlier:
    - save ~6 pm for large Monte Carlo production
    - + ~6 (?) calendar months for full analyses
  - invest man power in better and newer software to eventually benefit from larger flexibility
- cons:
  - will have to wait somewhat longer ( ~11 pm ) for first results
  - could speed things up by identifying more people to contribute

# scenario 3

- 'full program' for detector optimization with a large scale Monte Carlo production and a reasonably large set of physics benchmarks
- pros:
  - eventually will have solid understanding of the physics performance of alternative ILD detector models and tradeoff between cost optimization and physics reach
  - automatically includes scenario 2: will have first results sooner
- cons:
  - will have to wait some time for final results ( ~16 pm + X )
  - could speed things up by identifying more people to contribute

# Summary & Outlook

- DD4hep now has basic functionality implemented
- DDSim - new simulation package has first complete simulation model ILD_o1_v05 ported from Mokka
- **To Do:**
- Test and Debug new functionality and models
- interface to GEAR (backward compatibility of reco)
- finalise DDRec and port reco-software
- **Outlook**:
- apply new software for ILD detector optimization by defining 2-3 models and start investigating their performance (scenario 2 or 3)
- try to find people that can help so we can have the new models in the first half of 2015