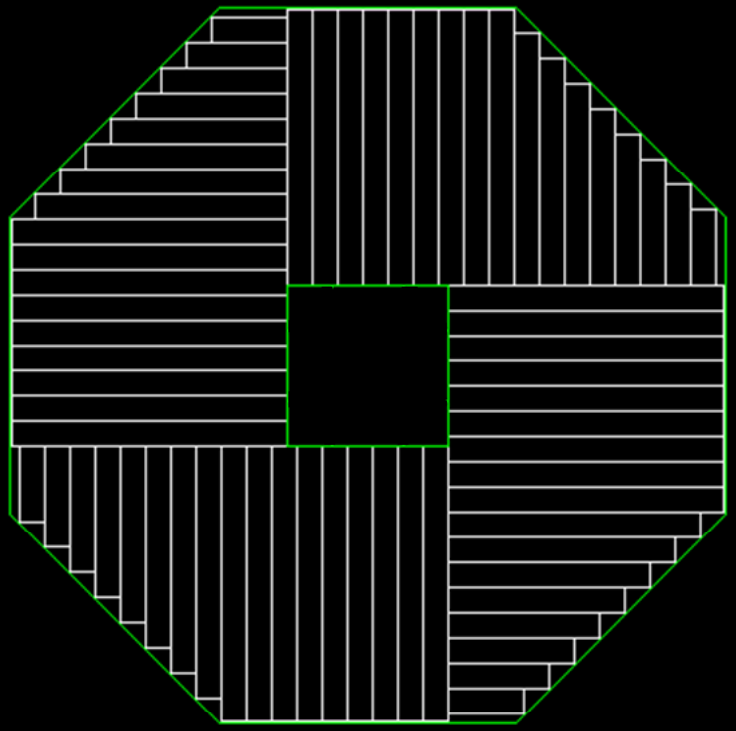
Status of Mokka



ILC Software Workshop at LAL – Orsay - May 2007

Paulo Mora de Freitas - L.L.R. – Ecole polytechnique

Talk plan

- Mokka overview and facilities
- What’s new since 6.0 (Cambridge)
- What’s coming soon
- Problems

Input : events

- Available interfaces :
  - Phytia HEPEvt (ASCII)
  - Phytia hepevt “Brahms-like” (ASCII)
  - Stdhep (binary)
  - Guinea Pig
- Particle gun :
  - Shoot individual particles
  - Possibility of smearing of position, direction and momentum

Output: hits

- Native ASCII format
  - A « Run » corresponds to a directory
  - Every sub-detector creates one or more hit files per event
- LCIO
  - Access to data via a high-level interface
  - API for C, C++, Java and f77
  - Automatically integrated to MARLIN and JAS

The simulation itself

- Full Geant4 based simulation
- Job controlled by a « steering file » and native commands of Geant4 and Mokka
- The user can choose among the Physics Lists supplied by Geant4:
  - We advice to use always xxx**_HP** (high-precision neutron models) in physics lists!
- User run time control on simulation, thanks to the User Actions pluggins

Plugin user actions

- Possibility of run « user actions » via plugin, at beginning/end of run, event, track or every step:
  - virtual void *BeginOfRunAction* (const G4Run *)
  - virtual void *EndOfRunAction* (const G4Run *)
  - virtual void *BeginOfEventAction* (const G4Event *)
  - virtual void *EndOfEventAction* (const G4Event *)
  - virtual void *PreUserTrackingAction* (const G4Track *)
  - virtual void *PostUserTrackingAction* (const G4Track *)
  - virtual void *UserSteppingAction* (const G4Step *)

Stable but flexible Geometry system

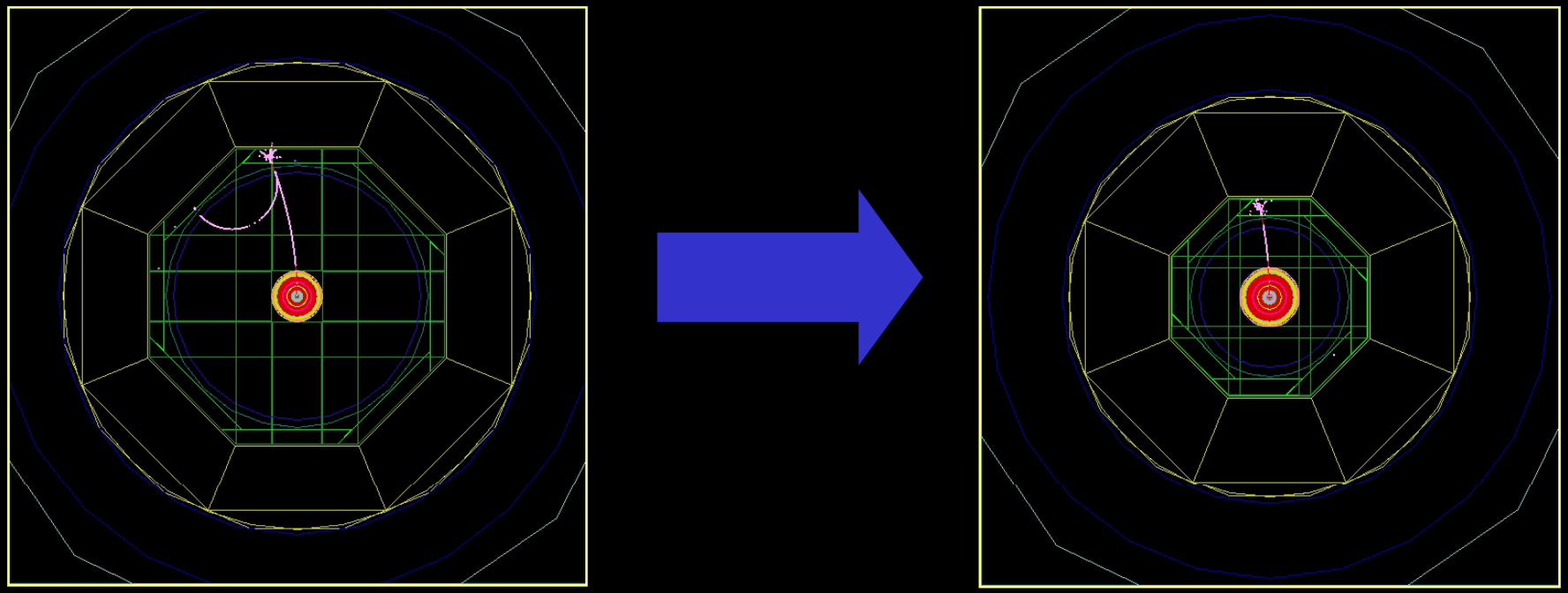
- Only one Geometry database (the Reference)
- But the user can easily:
  - clone the database and reuse the sub-detectors
- or, at launch time:
  - Re-scale the detector (*scaling models)*
  - And/or cook a new model
- Mokka provides also some facilities to modify interactively the model rendering (*visioning* models), for debugging.

« Scaling » a model (since aug05)

- The user is able to modify the model's main parameters at launch time, ex :

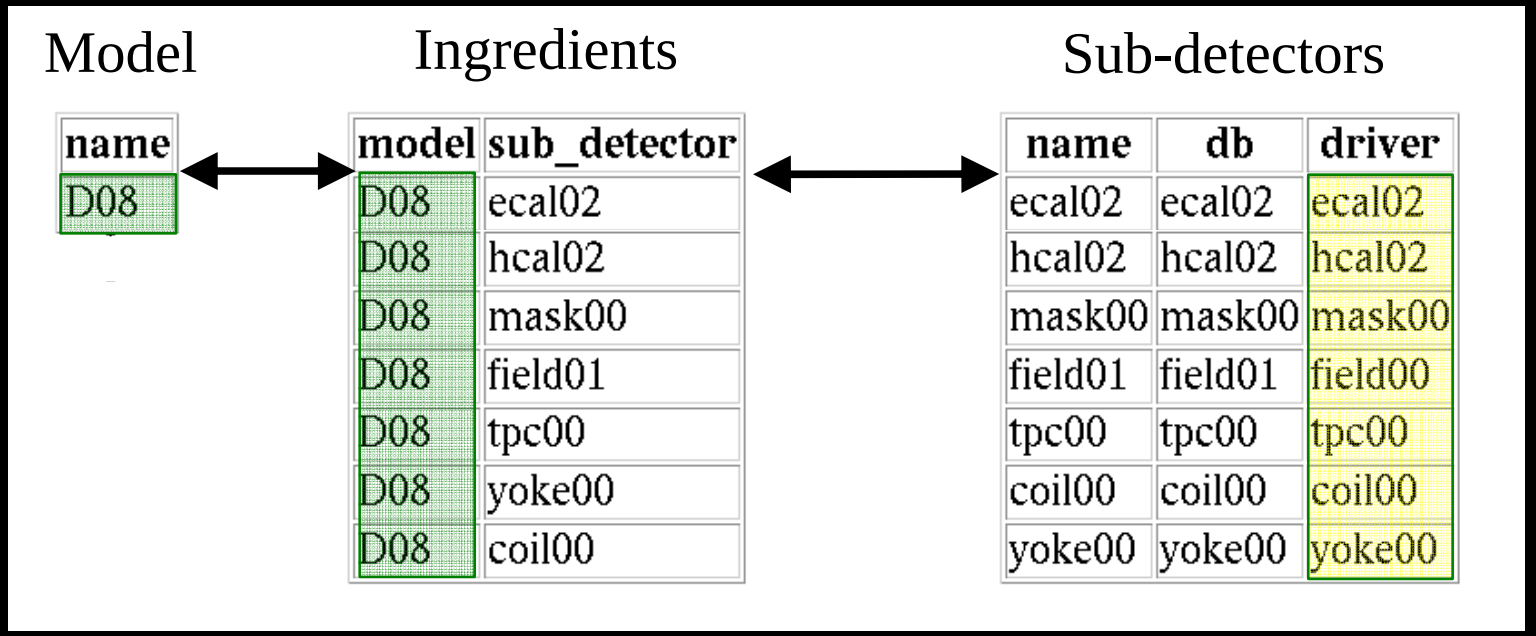
/Mokka/init/detectorModel LDC01

/Mokka/init/globalModelParameter TPC_outer_radius 800



Relational Geometry database

A detector model sample: “D08”



name
D08

model	sub_detector
D08	ecal02
D08	hcal02
D08	mask00
D08	field01
D08	tpc00
D08	yoke00
D08	coil00

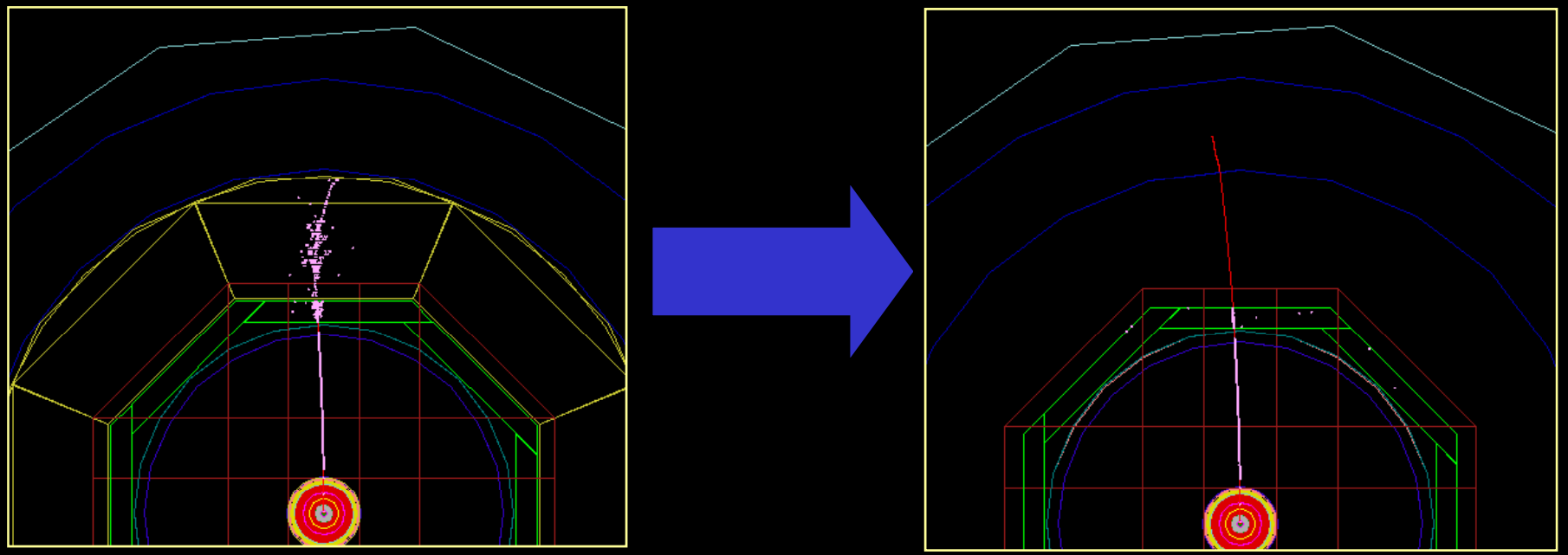
name	db	driver
ecal02	ecal02	ecal02
hcal02	hcal02	hcal02
mask00	mask00	mask00
field01	field01	field00
tpc00	tpc00	tpc00
coil00	coil00	coil00
yoke00	yoke00	yoke00

« Cooking models » (since jun06)

- The user is able to modify the model ingredients at launch time, ex :

  /Mokka/init/detectorModel LDC01

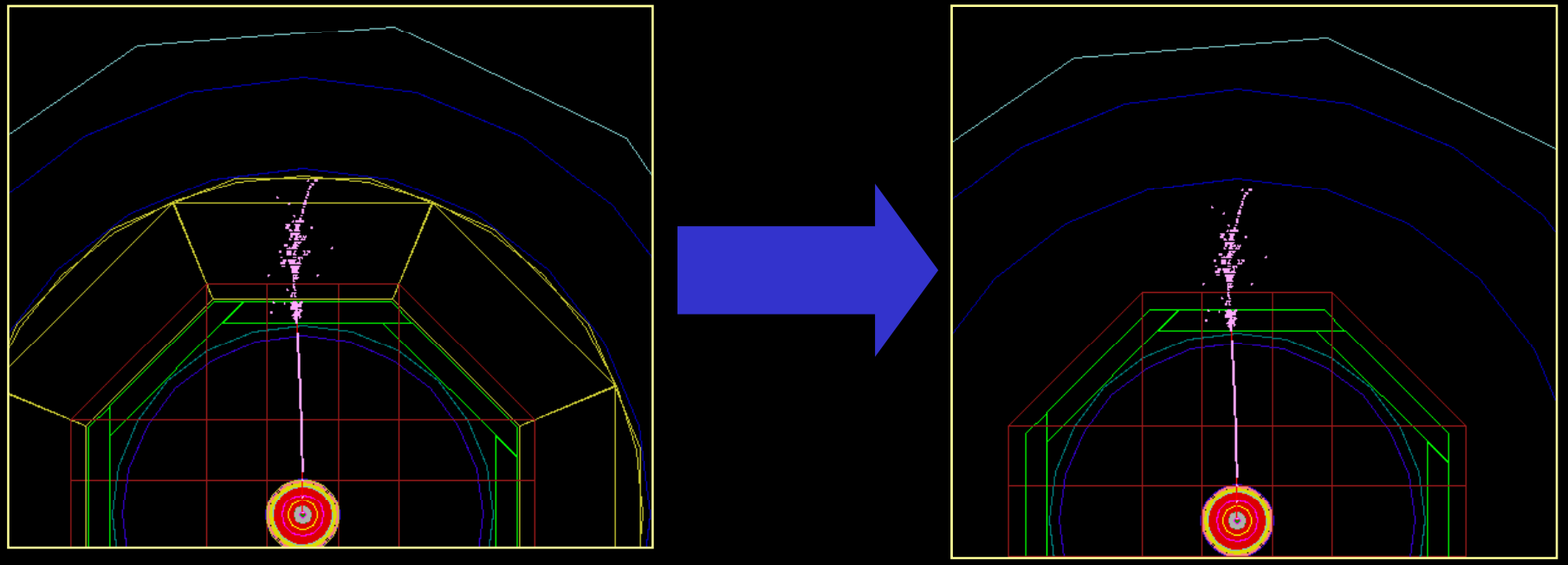
  /Mokka/init/EditGeometry/rmSubDetector SHcal01



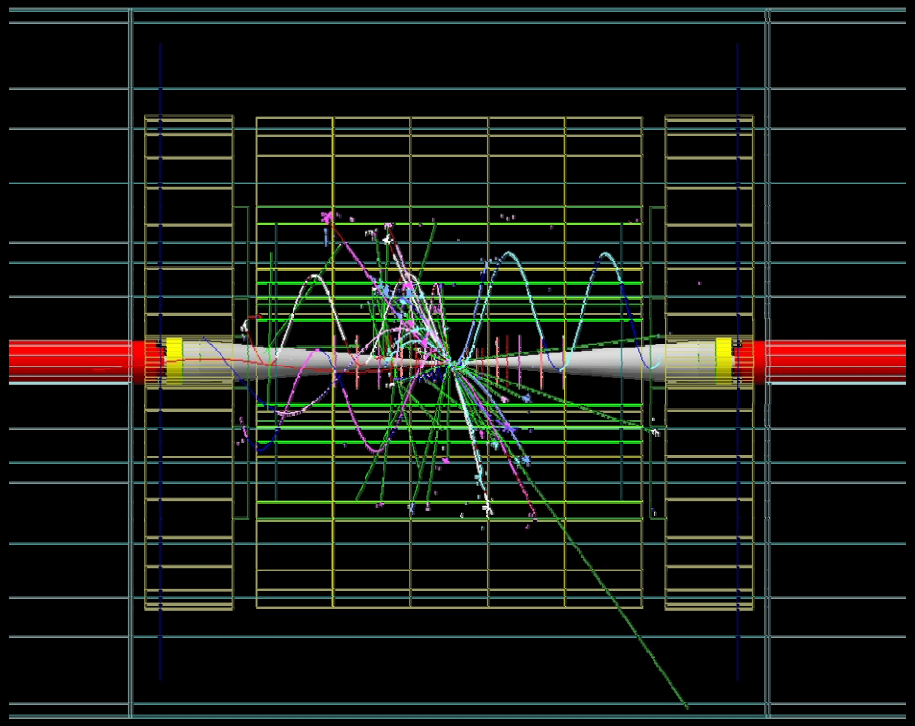
Visioning models (since mar06)

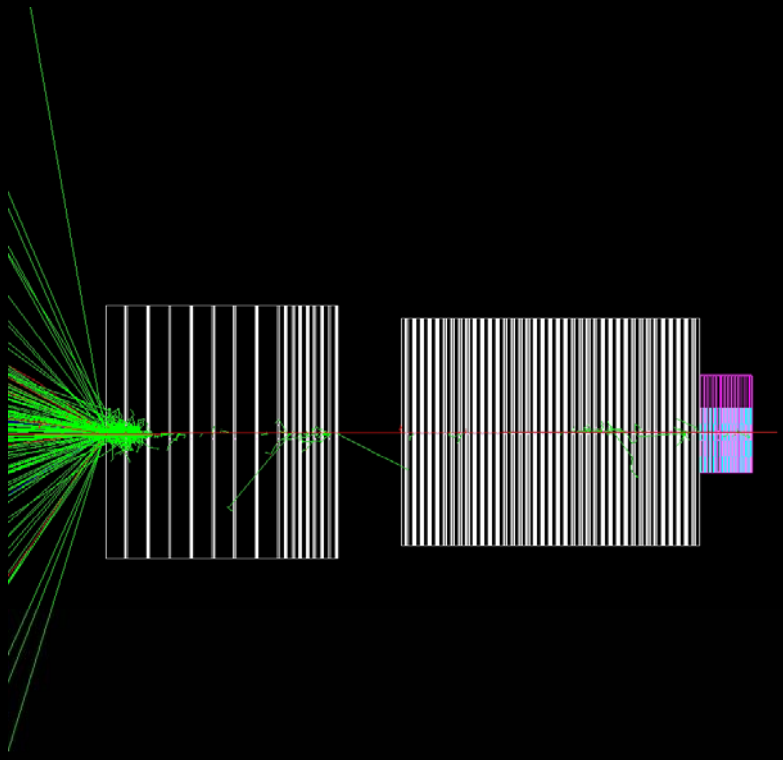
- The user is able to interactively modify the model rendering, ex :

  Idle> /Mokka/Visu/Detector/Visibility hcal false



Final detector and prototypes into the same simulation engine!





Test beams and « classic » models don’t scale

Model	Type	Obs. :
*D*xxaaa	baseline TDR	22 versions
*TB*aaa	Test beam	27 versions

where :

xx = “serial number” of the model

aaa = detail

Examples :

- “D12”, “D12scint”, “D14_CMOSVTX”,
- TBDesy0506_dchxy_new, TBCern1006_dchxy_new, etc.

Naming convention for new scaling models

- Model name = XXXYY_ZZtt where:
  - XXX = detector concept (LDC, SID, GLD)
  - YY = baseline:
    - 0 = Tesla TDR
    - 1 = LDC Detector Concept Studies
  - ZZ = release number (serial)
  - tt = Details (Rp = Hcal RPC, Sc = HCal Scintillator, etc.)
- Examples :
  - LDC00_02Sc, LDC01_02Sc, LDC00_03Rp, LDC00_03Sc, etc.

What's new since 6.0 - I

- Three main releases and two small patches since 6.0
  - Automatic creation of GEAR files
  - Mokka CGA API as the first GearPointProperties and GearDistanceProperties GEAR implementation
  - New Plugins
  - Several improvements concerning MySQL
  - Material Descriptions using G4NistManager
  - New steering command for initializing the seed of the random engine (/Mokka/init/randomSeed)

What's new since 6.0 - II

- Final Detector models:
  - SEcal01 with three different sets of radiator thickness
  - New detector models LDC00_02Sc, LDC01_02Sc:
    - new tpc scalable driver, EDT, Hcal and yoke improvements, realistic scalable vertex. (mainly Predrag, Frank, Adrien)
  - New detector models LDC00_03Rp, LDC00_03Sc:
    - Tube scalable driver with crossing angle support, LumiCal, Ftd and masks scalable drivers, etc. (mainly Bogdan Pawlik)

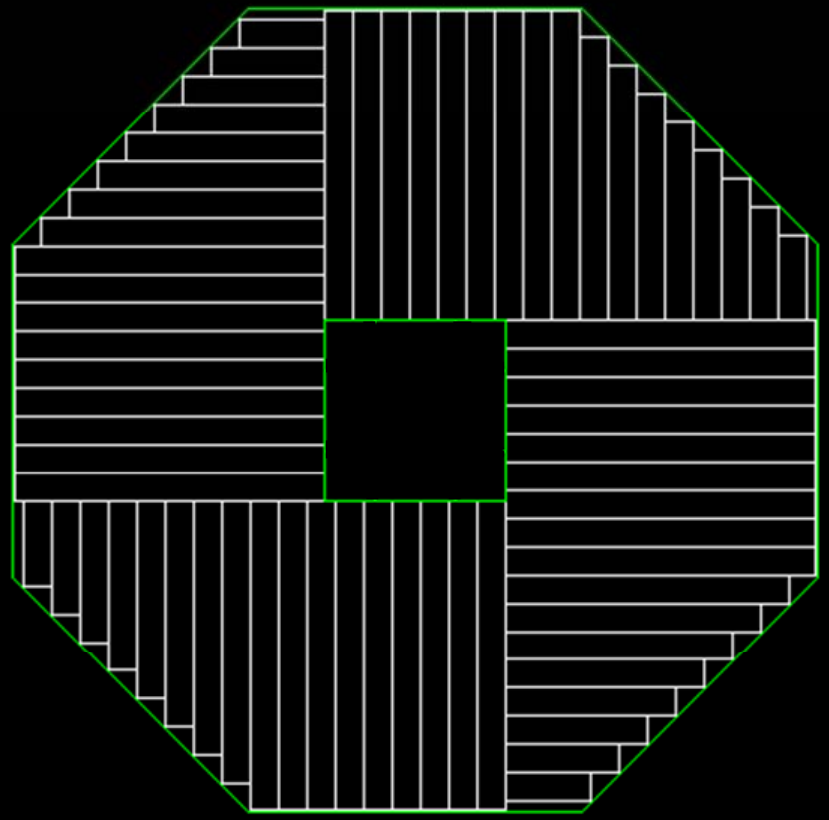
What's new since 6.0 - III

- A lot of job concerning the Test Beams:
  - New coordinate system for the Test Beam setups of 2006
  - Several ecal, drift chambers and scintillator new drivers and fixes for the DESY and CERN test beams
  - First implementation of DESY 5T magnet used for TPC prototype development (setup TM06)
- And several fixes for memory leaks, turn around for Geant4 bugs, etc…

Coming soon I : new SEcal03

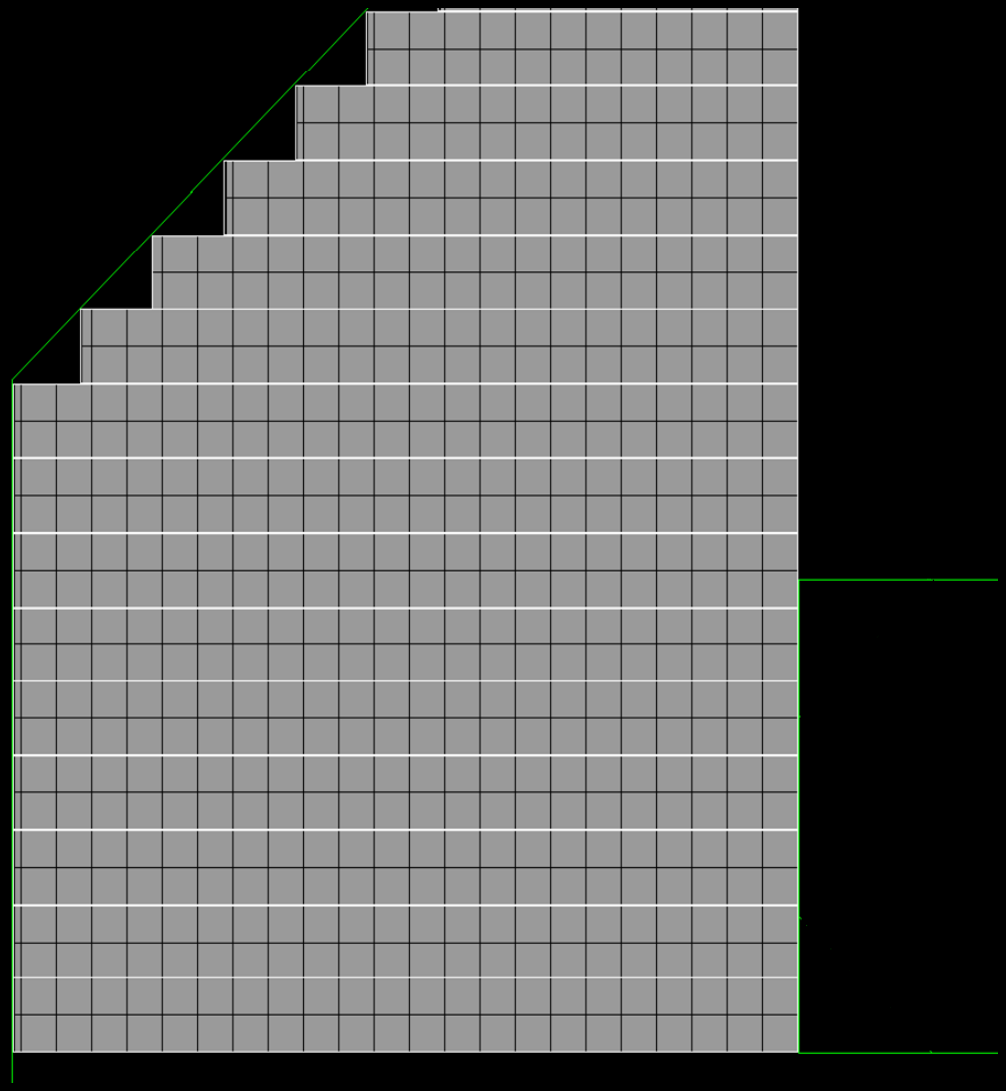
- Standalone scalable driver (no more creating temporary databases)
- A lot of run time parameters
- Implements “true” wafers, following the actual EUDET guidelines
- Odd number of layers 
- No more complicate Boolean shapes in end cap modules (should be faster) 

SEcal03: end cap modules



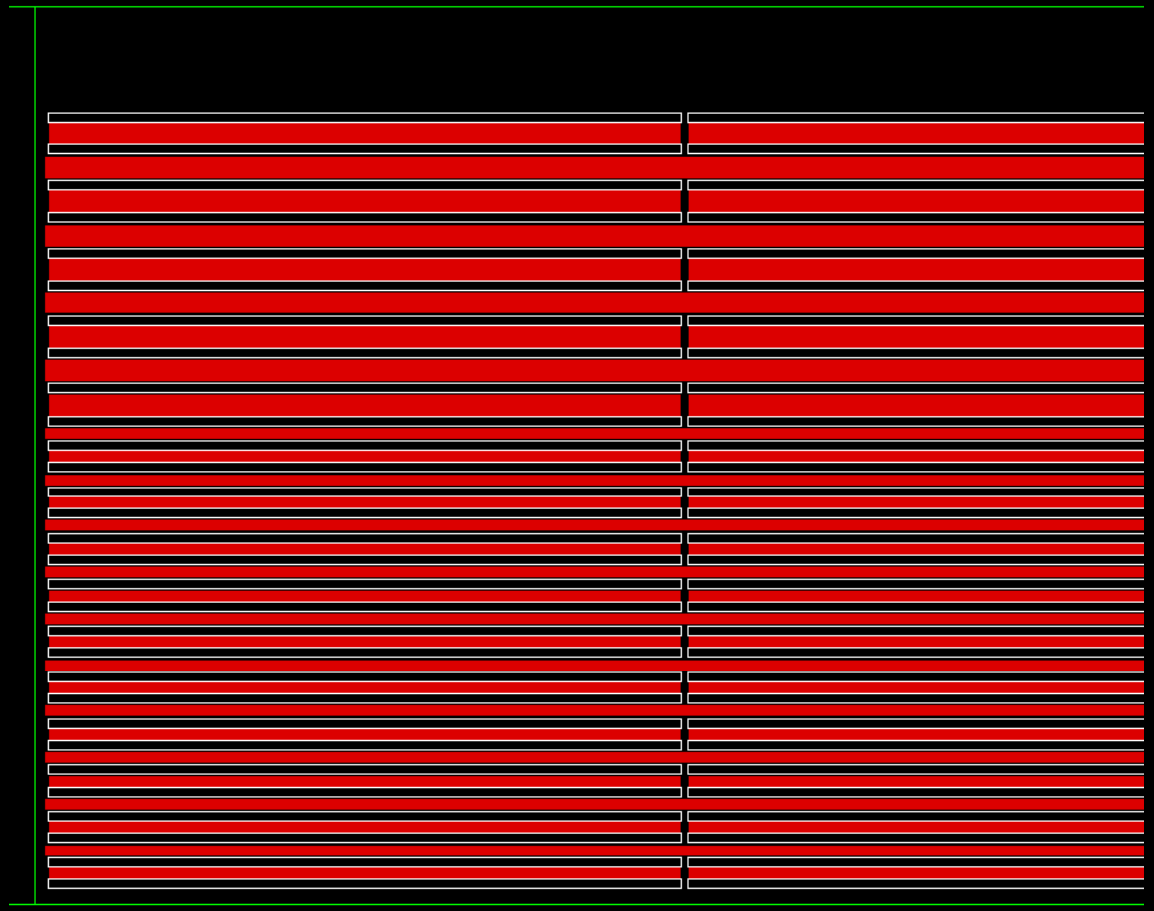
Paulo Mora de Freitas - L.L.R. – Ecole polytechnique

SEcal03: “true” wafers in slabs



Paulo Mora de Freitas - L.L.R. – Ecole polytechnique

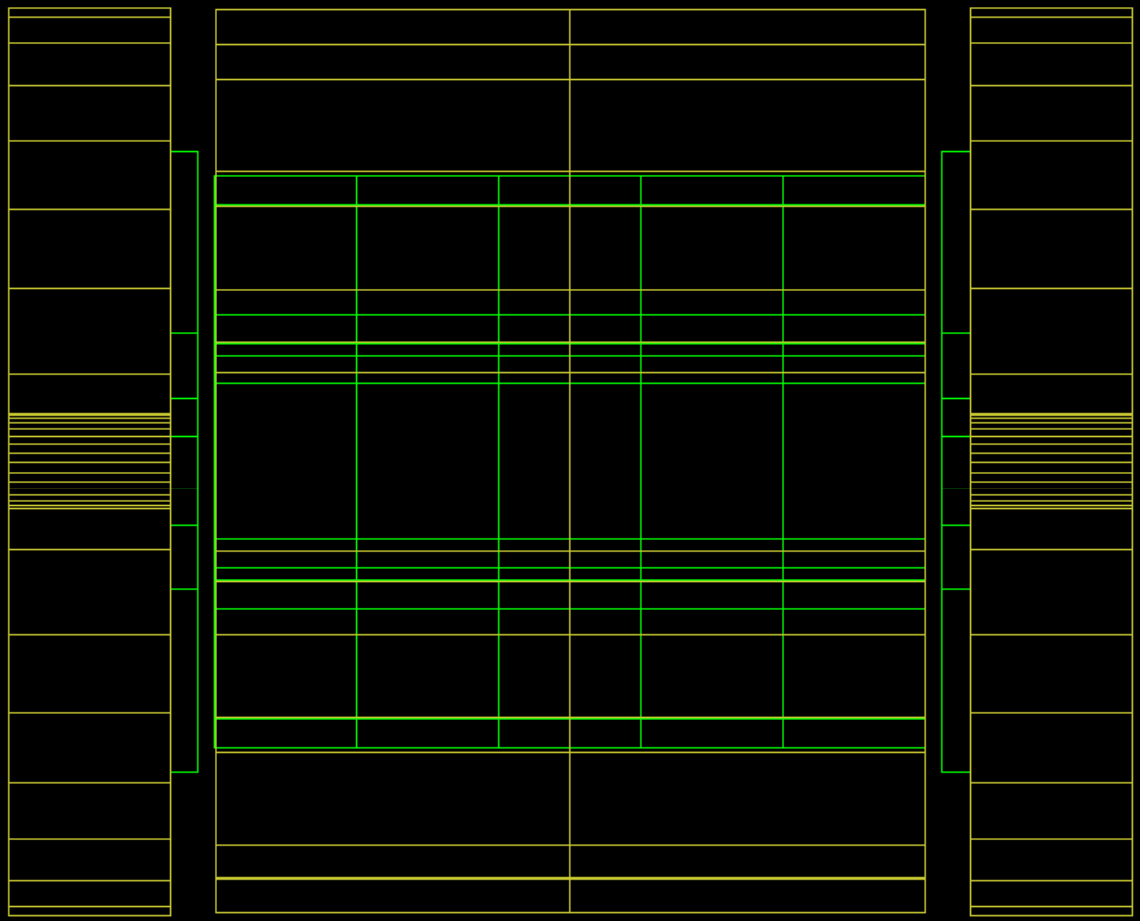
SEcal03, detail: fiber gap between slabs



Coming soon II : new SHcal03

- Standalone scalable driver (no more creating temporary databases)
- Almost the same as the actual LDC01 one (both scintillator and RPC modes)
- But
  - Just two modules per stave 
- Still missing: a decision concerning the Hcal rings, if one, where...

SEcal03 + SHcal03



Paulo Mora de Freitas - L.L.R. – Ecole polytechnique

So coming soon : a new DOD model

- We have almost all the pieces we need to compose it right now, but still missing:
  - Final details concerning Ecal, Si devices, etc.
  - Hcal rings, yes or not, where?
  - etc.
- To avoid a database inflation of intermediate models, could we still wait a bit ?

Main problems I: Documentation

- Only release notes are up to date:
  - No friendly documentation to get started
  - Mokka Web page is obsolete, missing descriptions of the current models in the DB
- Proposal for a solution:
  - To have concerned people helping to make up to date the Mokka site (it's Zope/Plone!)
  - To use automatic tools to retrieve information from database on the fly, to provide model descriptions via the web site

Main problems II: Coordination

- Works nice for the Test Beams developments, thanks to the CALICE structure

BUT

- Missing for the final detector models:
  - Who decides what should be the next DOD model?
  - Who is the responsible for each sub-detector part in simulation?

Thanks a lot!

- Mokka became *de facto* a very nice, friendly informal collaboration
- Many thanks for a lot of people helping to develop and to maintain it

Please, fell free to join… and to enjoy it!