GEANT 4 SUPER*B* SIMULATION: OPEN ISSUES & WEAKNESS

Lessons Learned:

or

How we Will Survive From Here to the TDR "the End"

VOLUME HIERARCHIES

- In Geant4 the geometry tree must meet these requirements
  - Physical volumes placed inside the same logical volume cannot overlap (even if made of vacuum)
  - Physical volumes places inside a logical volume cannot protrudes from the mother volume
- If these requirements are not met the "Geant4 results are unpredictable"
  - So far BaBar geometries protected us from clashes: we have to decide what space belong to who.
    - Each subdetector should present a proposal of the shape and size of its envelope.

GDML BUG HUNTING I

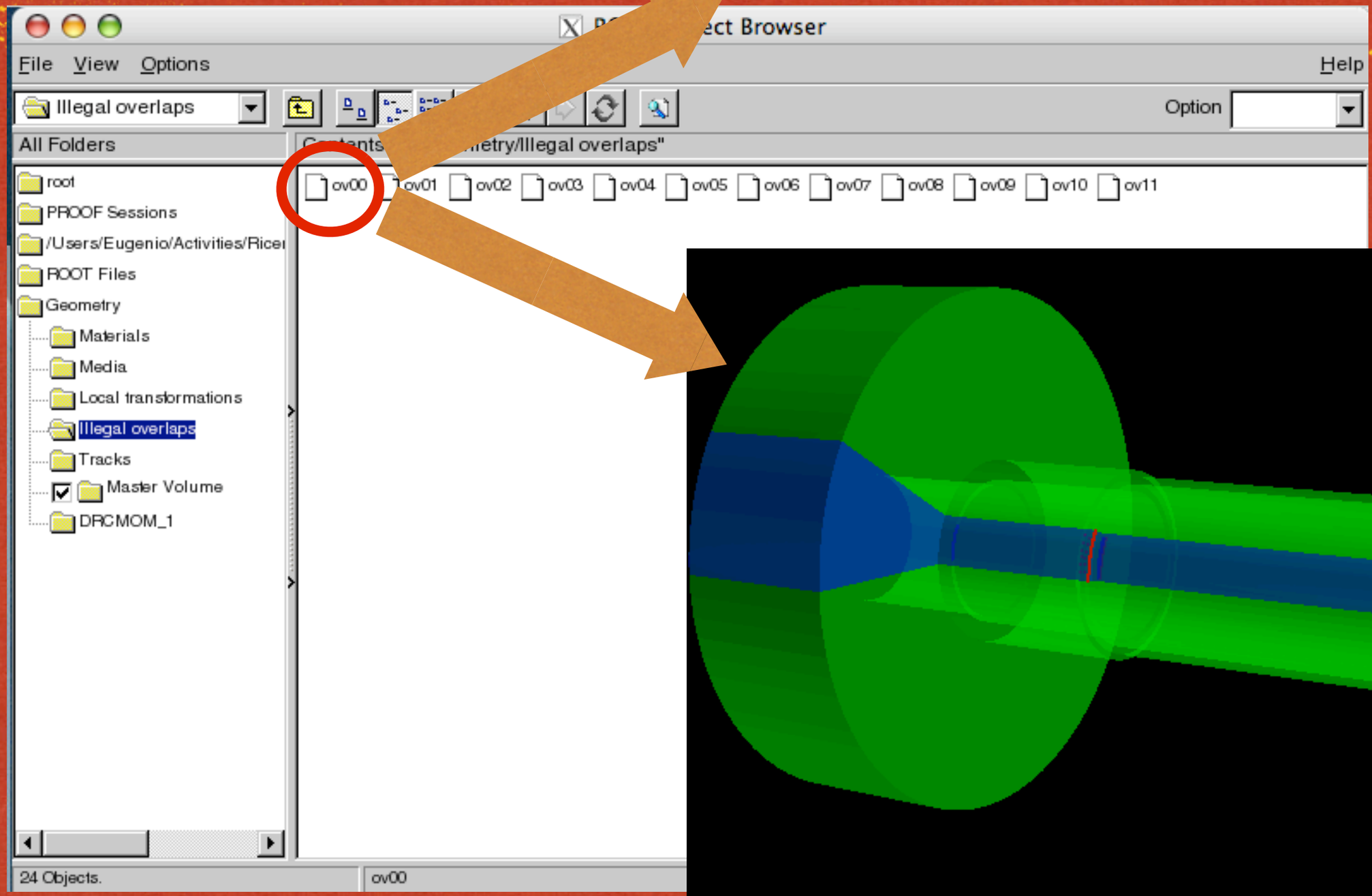
```
Compiled on 30 May 2007 for macosx with thread suppor

CINT/ROOT C/C++ Interpreter version 5.16.19, Ma
Type ? for help. Commands must be C++ statement
Enclose multiple statements between { }.
root [0] gSystem->Load("libGeom");
root [1] TGeoManager::Import("DRCMOM.gdml")
Info: TGeoManager::Import : Reading geometry from file: DRCMOM.gdml
Info in <TGeoManager::TGeoManager>: Geometry Geometry, default geometry created
Info in <TGeoManager::SetTopVolume>: Top volume is DRCMOM. Master volume is DRCMOM
Info in <TGeoManager::CheckGeometry>: Fixing runtime shapes...
Info in <TGeoManager::CheckGeometry>: ...Nothing to fix
Info in <TGeoManager::CloseGeometry>: Counting nodes...
Info in <TGeoManager::Voxelize>: Voxelizing...
Info in <TGeoManager::CloseGeometry>: Building cache...
Info in <TGeoManager::BuildCache>: --- Maximum geometry dep
Info in <TGeoManager::CloseGeometry>: 709 nodes/ 16 volume U
Info in <TGeoManager::CloseGeometry>: ------------modeler ready---
(class TGeoManager*)0x202dc00
root [2] gGeoManager->GetTopVolume()->CheckOverlaps(.01)
Info in <TGeoVolume::CheckOverlaps>: === Checking overlaps vor volume DRCMOM ===

Info in <TGeoVolume::CheckOverlaps>: Number of illegal overlaps/extrusions : 12
```

Import libraries then the geometry

Check for overlaps/protrusion with an accuracy of 0.01 mm

OVERLAPPINGS/PROTRUSION

```
root [4] new TBrowser
(class TBrowser*)0x70b3710
root [5] <TCanvas::MakeDefCanvas>: created default TCanvas with name c1
 = Overlap ov00: DRCMOM extruded by: DRCMOM/DrcSec_59 ovlp=6.914
```

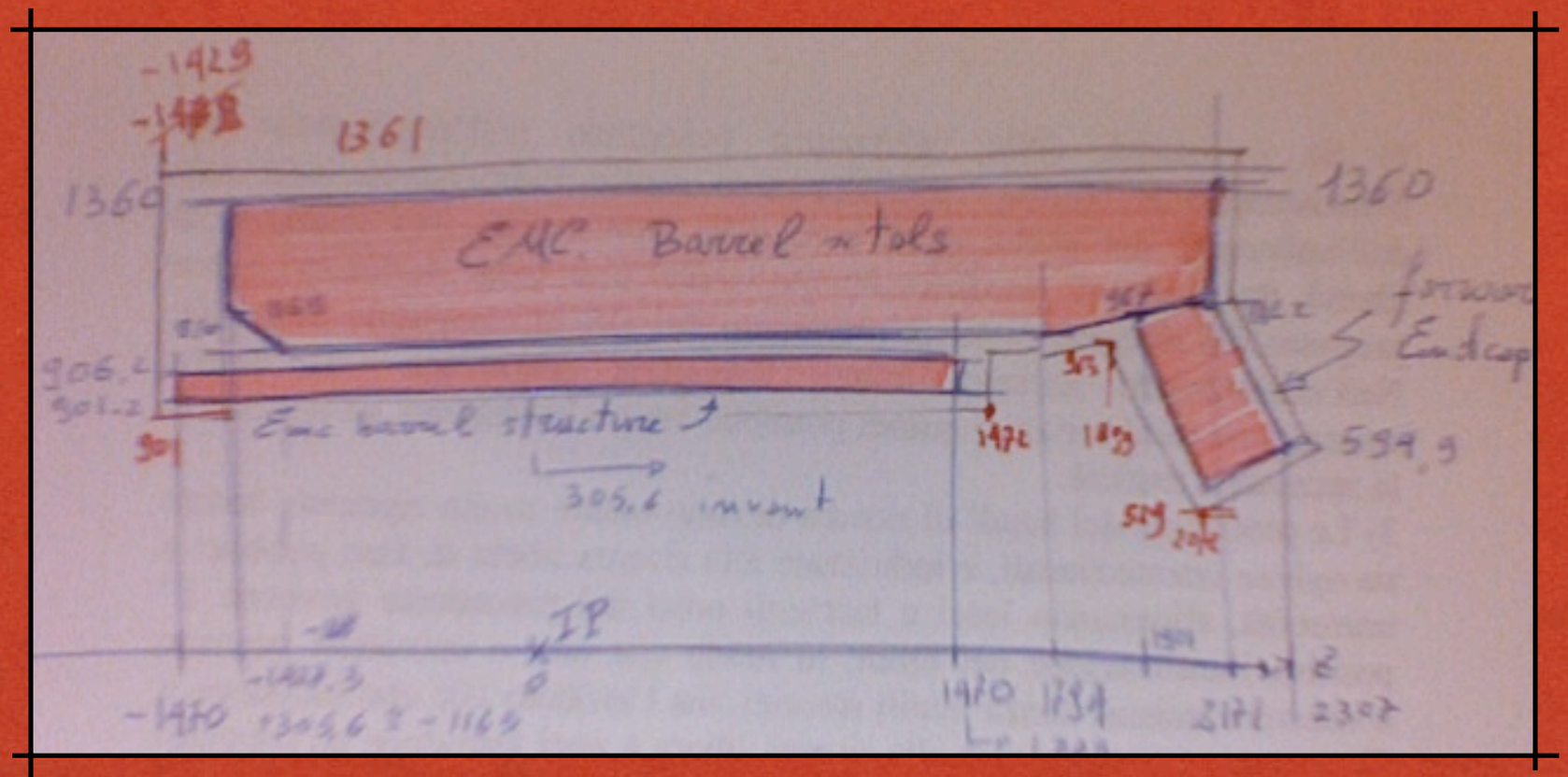


GDML BUG HUNTING II

- Several GDML parser: GDML libraries, root interpreter, next minor upgrade of Geant4
- Each parser is in subtles ways different from the other ones: last sentence left to the parser used in the simulation. E.G.: root 5.15.08 does not understand the IFR_FWD_ENDCAP gdml, simulation understand it, root 5.15.08 is wrong.
- The simulation program is able to read the GDML and write a simpler one with expression expanded: Bruno -g input.gdml -o out.gdml

DOCUMENTATION

- We are defining geometries: we have to draw it. How? Pencil on paper? Power-Point? Xfig? Inkscape? We have to decide it by the end of the meeting



IS IT SO URGENT?

Re: MC Ks decay vertices ?!?

Base: Preliminary/unconfirmed Bugs, Problems, Frustrations, Fixes
Re: MC Ks decay vertices ?!? (Marcella Bona)
Keywords: MC ks decays
Date: Fri, 30 Jul 1999 23:55:23 GMT
From: Marcella Bona <bona@slac.stanford.edu>

Hi all!

Those MC ks are in the MC list and they do decay in the [19<r<23]cm region but the daughter-mother info is lost! This should happen because the [19<r<23]cm volume is considered by BBSIM to be BFAC and not Svt or Dch: as a consequence the Geant information on the decay is not stored.

So that the Ks can be reconstructed in this volume, but the Geant3 truth info is not available. As a matter of fact we have found in the MC list some Ks with no daughters!

(on reco Ks, see

http://www.slac.stanford.edu/~bona/ks/gapRec1.gif

http://www.slac.stanford.edu/~bona/ks/gapRec2.gif)

Thanks to all!
Ciao

marcella & eugenio

BaBar

This bug in the geometry was unnoticed for years before being discovered and solved

SENSITIVIZATION

- At present the sensitivization is a kludge™
- C++ code navigate trough the geometry tree and sensitivize the volumes with given materials (EmcCsI, EmcLSO, SvtActiveSilicon, WiGa, IFR_SCINT_MAT etc) or with given names
- We have to design GDML attributes to keep the C++ code free from material_name, volume_name etc. etc. how? Volname convention? extra GDML tags?

SEGMENTATION

- The native GDML does not capture the relation among a given volume and the read-out channel name: temporary kludge solution developed in C++ for the SVT (very clumpsy) and for the EMC (numebering scheme obscure) so far

- What indexing scheme do you need? How many indexes? Can the index can be attached to the volume name? to the position name? to the copy number?

EMC BARREL PHI INDEX

```
virtual G4int GetIndex(G4Step* aStep){
        G4StepPoint* preStep = aStep->GetPreStepPoint();
        G4TouchableHistory* th = (G4TouchableHistory*)(preStep->GetTouchable());


        #ifdef VERBOSE_OPERATION
        for(int i = 0 ; i< 6; i++){
                G4cout << th->GetReplicaNumber(i) << "\t";
        }
        G4cout << G4endl;
        #endif

        G4int EBMX_index = th->GetReplicaNumber(2);
        G4int EBAP_index = th->GetReplicaNumber(4);
        return EBMX_index + EBAP_index*3 + 1;
}
```

Barrel

The algorithm
is not resilient to geometry
hierarchy changes

Endcap

```
G4int CF_COLUMN_index = th->GetReplicaNumber(1);
G4int CF_MODULE_index = th->GetReplicaNumber(2);
return  CF_COLUMN_index + 5*CF_MODULE_index + 1;
```

SVT WAFER INDEX

```
newHit->SetLayer( VolumeName(2)-'0' ); // What a HACK!!
```

- To set correctly the PhysicalVolumeName was necessary to modify the GDML library...
- Recipe: give the PhysicalVolumeName according to the position ref name
  - It works only for “flat” hierachies: i.e. SVT, IFR, (not for EMC)

VALIDATION

- Background simulations are not ideal environments to understand detector responses:
  do you need a specific sample to simulate to validate the simulation? xxx single particles events with this and that energies in this angular region... as an example.
- Other kind of check?

DIGITIZATIONS:

- Do you need extra informations?
- Path to the digitization options
  - Digitization inside Geant4 framework
    - Pro: fast development time
    - Cons: one shot, we will have to rewrite the code when we will have a BaBar tcl-like framework
  - Digitization outside Geant4 framework
    - Pro: multiple shot (fail the first? try again)
    - Cons: disk space, framework, hit persistency

CONCLUSIONS

- Bruno (after Bruno Touschek, father of storage rings) have still to be improved IFR segmentation etc. etc.

- in the short term it can be used (after proper validation) to simulate backgrounds/physics event with the present detector and with slight modified versions of it

- but... on a longer time scale we will have to make some rework on top of it.