MEMPHYS on Lund/iridium MEMPHYS_vis and... what next ?

Uppsala / 1 October 2017

MEMPHYS

- MEMPHYS for... MEgaton Mass PHYSics !
- Geant4 simulation of a large scale water cherenkov detector.
- Primary author of the code ? M.Fechner ?
- No license and clear authorship = public domain!
- At LAL, around 2005, used and improved by Jean-Eric Campagne and Nikos Vassilopoulos.

MEMPHYS in my life

- Involved as Geant4 support at LAL. I was and I am still an active Geant4 member (analysis tools, visualisation).
- This simulation helped to test and debug the « optical » and Cherenkov processes that was rather new in Geant4 at that time (P.Gumplinger, TRIUMF).
- Contribution to store a minimal « event model » in a file at the ROOT format but by using my very light Rio package through an « AIDA » implementation (long story...).
- Contribution to provide a 3D display for MEMPHYS based on Coin3D for the graphics and my OnX package for the GUI.

MEMPHYS hibernates

- Group neutrino at LAL => cosmology (Planck, now LSST).
- MEMPHYS hibernates and the world changes !

Wake up begin of 2017

- Request from Nikos to wake up MEMPHYS.

Done (1) @ LAL

- At CC/IN2P3, system, compilers, geant4 and all that had changed.
- Have it built and run on my Mac with the original geant4-8.2 to be sure of the physics before changing things. (Built with Apple/clang)
- CRITICAL : update the code to run on recent geant4-10.x.
- « DetectorConstruction.cpp » not touched, but had to update the « PhysicsList.cpp ». It looks ok but have to be cross checked by someone close of the physics.
- Modifs are « #ifdef GEANT4_VERSION » so that the code can still run on the geant4-8.2 !

Done (2) @ LAL

- Had to change the build system (CMT no more maintained). Use something (« bush ») much more simple based on bash (long story). (Rustic, but it permits me to build also for Windows, Android, iOS). But could have the geant4 cmake example logic for laptops.
- Had to change the way to store the « event model » (AIDA RIP around 2010). Use now my inlib/exlib for that. Very light code to store histo & ntuple at the root format without having to embark/tie to the whole CERN-ROOT for that. Now coming in geant4/analysis tools with the « g4tools ». (long story too).
- Have to build with c++11. Ok.

Done (3) @ Lund

- Account on iridium in May, thanks to Florido Paganelli.
- Install first the geant4-8.2 version by using the default compiler : no resistance. Run a simple slurm job, ok too.
- Geant4-10, needed more work :
  - Default compiler not c++11 compatible, find and use a 4.9.3 from /cvmfs/sft.cern.ch/lcg
  - Have to install a recent cmake (3.8.2) needed by Geant4-10.
  - Build geant4-10. no problem here.
  - Build and install MEMPHYS/geant4-10. It works ! ☺

Done (4) @ Lund

- Install clang (3.4). Thanks to Florido.
- The idea is to use a local compiler and have a full local build (it may ease a lot debugging).
- Build geant4-10 with it. (In fact, I think it is probably the first G4 app built with Linux/clang++/c++11 !).
- A slight resistance with one Geant4 file => reported.
- But MEMPHYS works with it ! ☺

Done (5)

- Right now things readable (and copyable) at :

  /nfs/users/gbarrand/usr/local and ~/public
- Have a git repository on github :

  https://github.com/gbarrand/MEMPHYS.git
- Two releases/branches :
  - MEMPHYS / version_9_0_0 : on geant4-8.2
  - MEMPHYS / version_10_0_0 : on geant4-10 with g++
- There are README files to build, install and run.
- Have to do a release/branch for geant4-10 and clang.

Not done : look parallelism

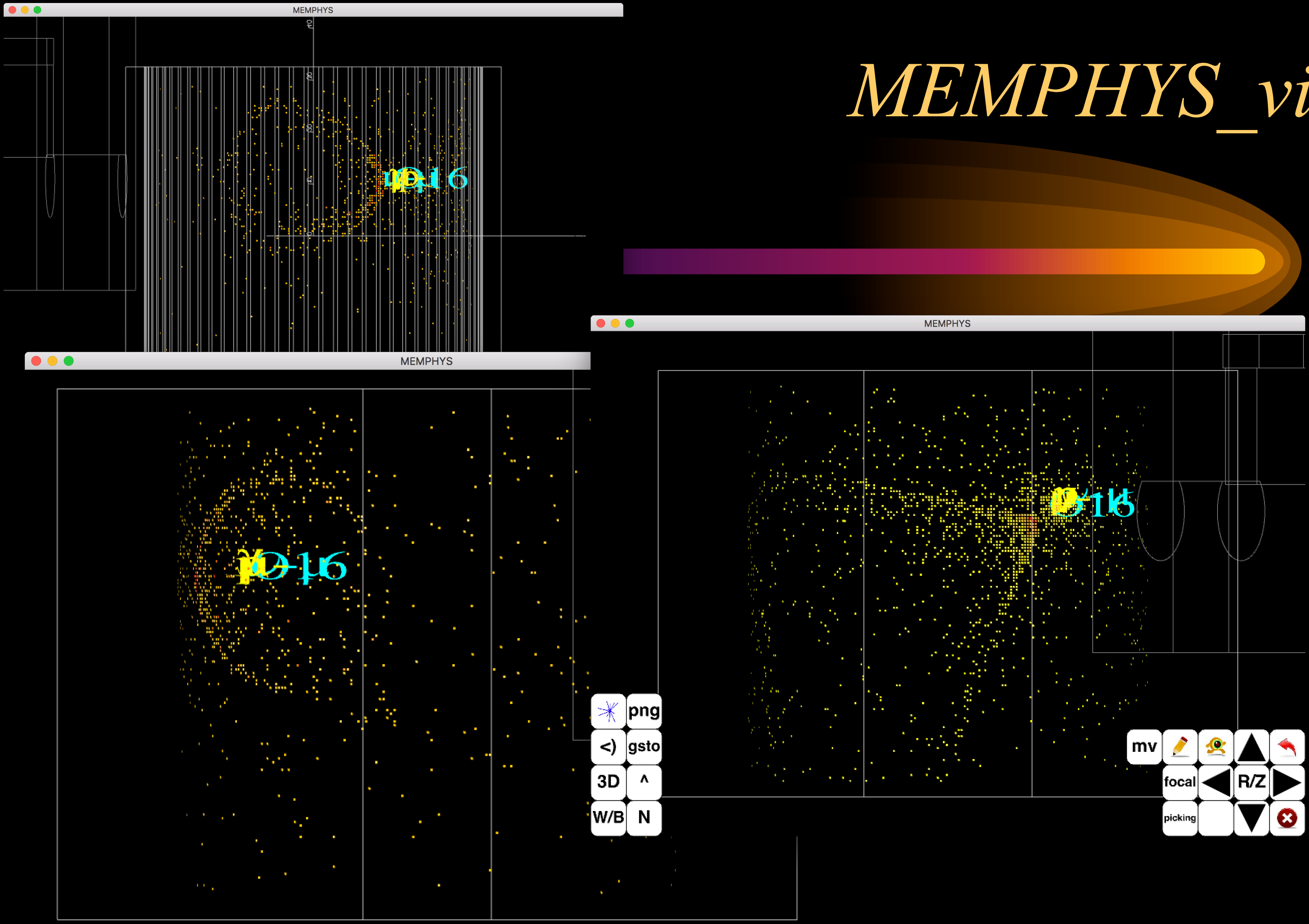
- Support of multi-threading is available in the geant4-10.x.
- Around IO in G4analysis category, there is now ways to gather histos and ntuples with same booking coming from multiple threads or MPI/workers within one main file.
- Gathering output is a problem if having a lot of parallel units. If there is no MPI on Iridium, we may look for another mechanism to do that (we have code now to pass histos and ntuples/pages in messages).

Restore some visualization MEMPHYS_vis

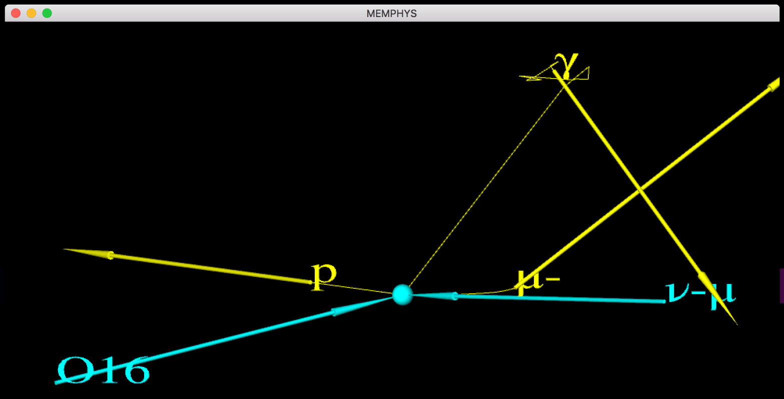
The world changed...

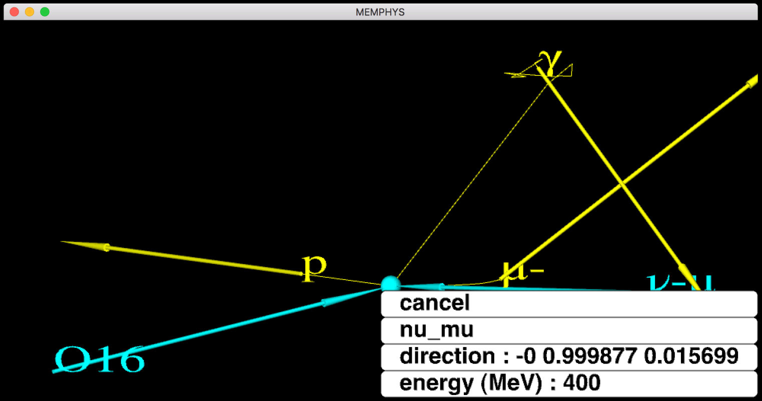
- 2007 iPhone, 2010 iPad and then the Android world with similar devices (really ? ☺) => A lot of CPU and graphical power in the hands of people !
- I want to deal with that => inlib/exlib around 2011. See my « softinex » portal describing all these.
- July : MEMPHYS/10.1.0 with a first version of MEMPHYS_vis that uses my inlib/exlib (C++ & GL-ES) code to do visualization.
- Installation on the euronunet virtual machine.

MEMPHYS_vis



MEMPHYS_vis



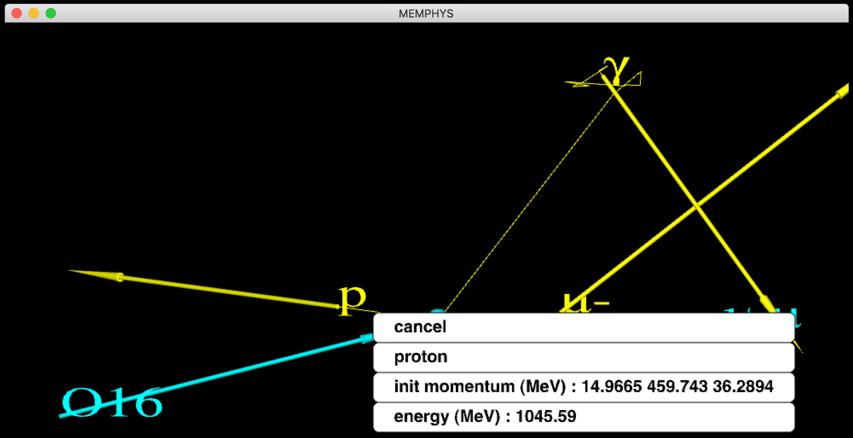


cancel

nu_mu

direction : -0 0.999877 0.015699

energy (MeV) : 400

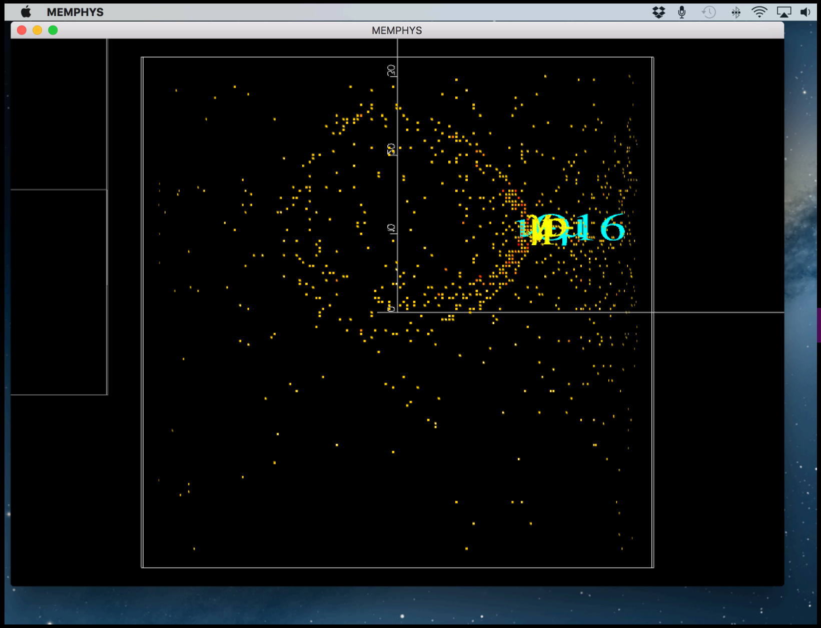


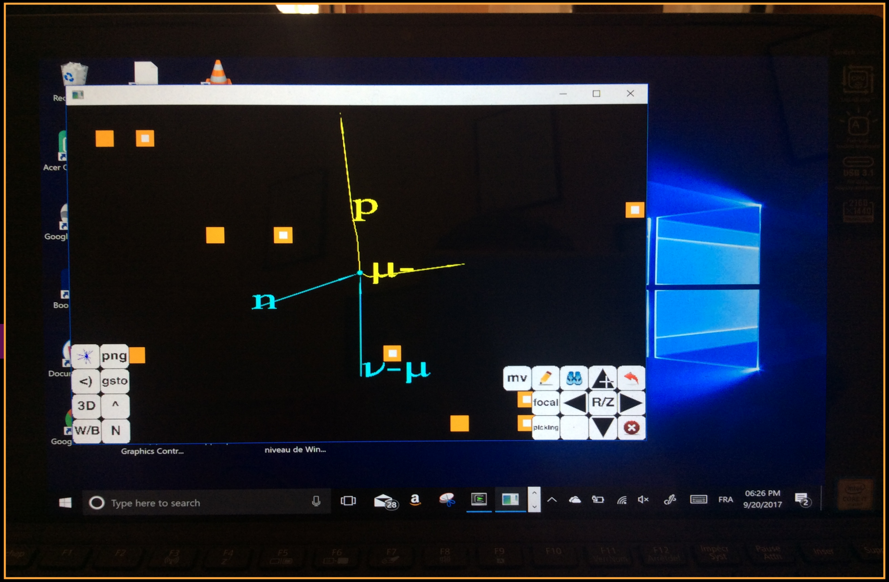
cancel

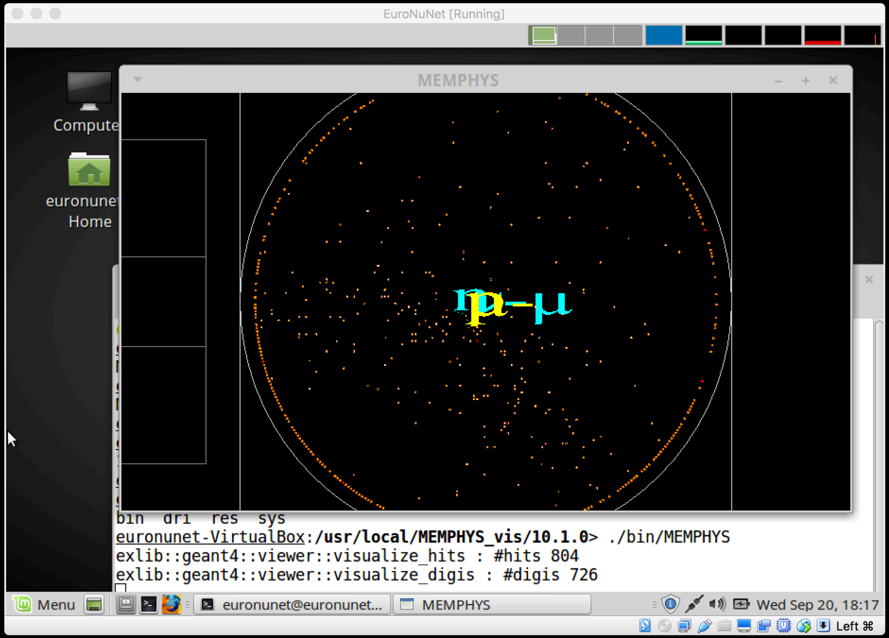
proton

init momentum (MeV) : 14.9665 459.743 36.2894

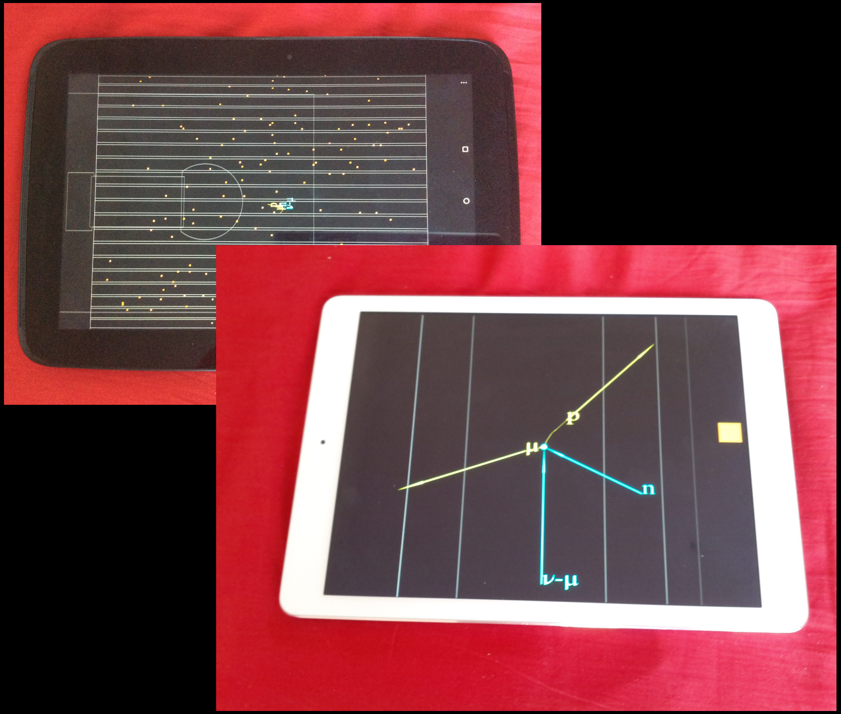
energy (MeV) : 1045.59







```
bin  dri  res  sys
euronunet-VirtualBox:/usr/local/MEMPHYS_vis/10.1.0> ./bin/MEMPHYS
exlib::geant4::viewer::visualize_hits : #hits 804
exlib::geant4::viewer::visualize_digis : #digis 726
```



Today MEMPHYS_vis is unique !

- Probably the only G4 application with a visualisation running on the five today interactive and « close to people » environments (Linux, macOS, Windows, iOS, Android).
- (G4 and ROOT vis systems can’t do that).
- (And if we include the fact to be able to write/read data at the ROOT format in a light way on all these devices, it is definitely unique in HEP).

Some doc

- I have some README to build and run coming from the packages.
- There is a MEMPHYS section under my portal :

  http://softinex.lal.in2p3.fr
- (I have to put some specific doc to build MEMPHYS_vis from source).
- There is a lot of presentations about the inlib/exlib « way of thinnking ».
- There is a demo of MEMPHYS_vis on YouTube.

Next…

NOT DONE. Some g4euronu ?

- Install/deploy on a more public place under iridium.
- Why ? Because MEMPHYS is the past of another detector that does not match what you want to do. If developing for a new one, you have to deploy for the new one.
- You have to clone MEMPHYS for your detector, your physics and your development environment !
- In the code, « MEMPHYS » appears mainly in « namespace MEMPHYS » which is easy to chamge (for g4euronu, or something).
- (I can help to migrate...)

HEP software and today interactive technologies

Today HEP software ☹

- There are today « deep » issues around HEP software.
- Beside probably no way to gain a factor 10/100 around the « batch » (reconstruction, simulation, analysis), right now HEP is passing aside a whole trend in interactive technologies : smartphones, tablets, VR, AR, wall of screens.
- Here the technology is pushing hard (see specs of the Note-8 and the further iPhone-X(pensive)), and HEP is left behind.

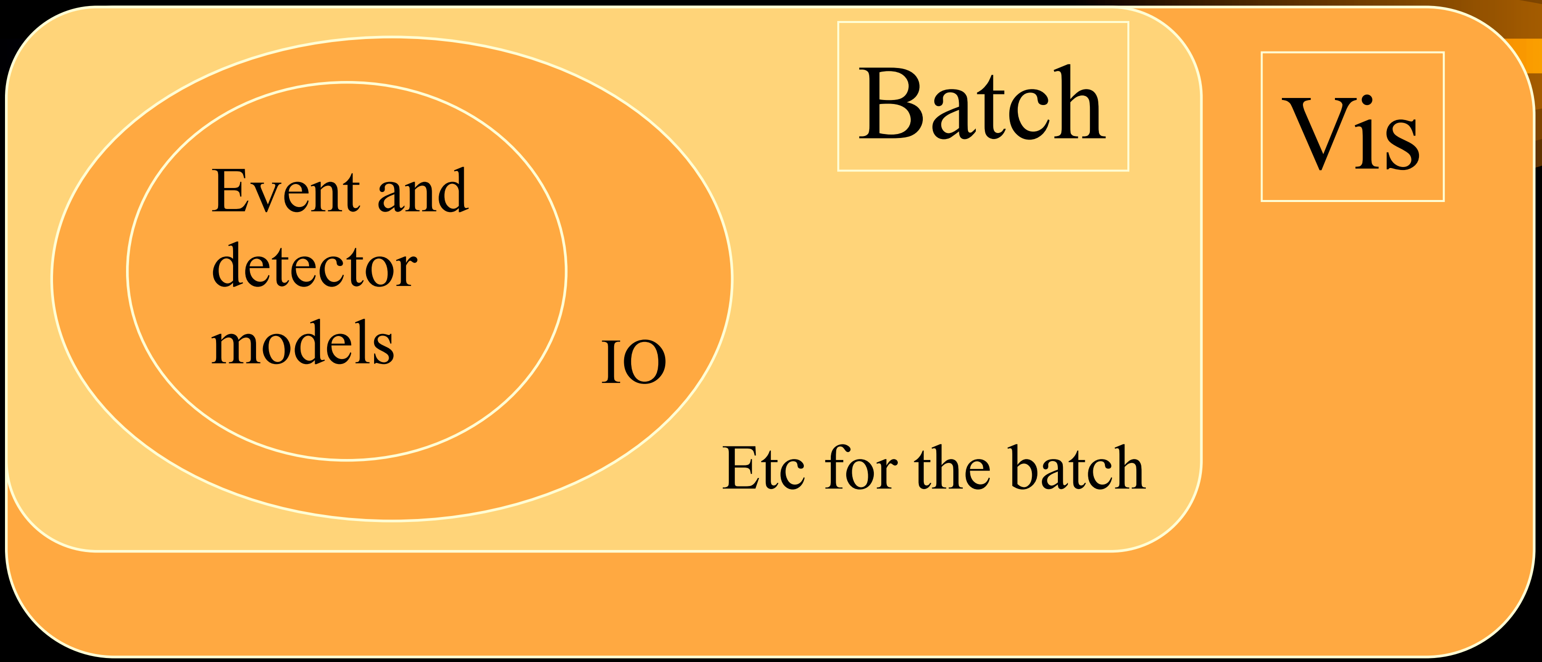
Today HEP software ☹ ☹

- At least for outreach, HEP is missing an historical and unique occasion to put its physics in the hands of people !
- People that... pay our accelerators and experiments !

HEP software ☹ ☹ ☹

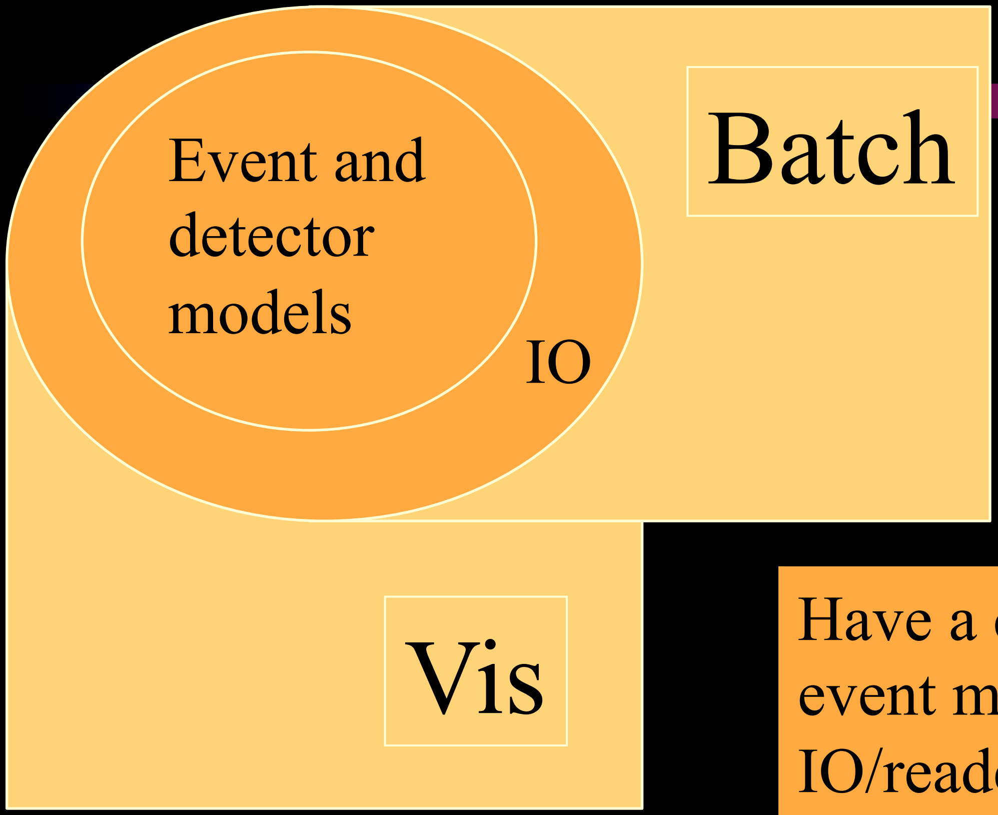
- Around the LHC, there is no way right now to open an event file and exploit it on local devices « close to people ».
- Because today software came from some kind of « code inflation » around some « learning code kernel » that is today no more portable and no more evolutive.
- Today moto at CERN for handling interactivity : hide the whole thing in servers and do the graphics, plotting through the web. It may work, but it is definitely not the optimum.
- We loose high « reactivity » for the interactivity.

Today = "batch oriented"



- Event and detector models + IO immersed in a no more portable environment.
- Vizzers have to build on the « batch » software => stuck

Future = “batch & vis oriented”



Event and detector models

IO

Batch

Vis

Have a common highly portable event model, detector model and IO/reader

Be able to read data everywhere !

Why coming at Uppsala ? Involvement ?

- There is no hope around CERN and HSF.
- Around HEP, I am looking for a new program/experiment for which « software people » would share such a view !
- Not only we would gain around interactive technologies but, why not, « see » new ways to boost the batch...

- In MEMPHYS there is already an ambryo of an event model + IO highly portable. (And I can also read a geometry from a .root file). It may be used as a seed for a new generation of HEP software...

Is there some echo ? ☺