



ALICE OFFLINE :ORGANIZATION

A (PHOS biased) user point of view

Yves Schutz, SUBATECH, France

Toward AliRoot ...

A **single** framework for **all** computing activities in ALICE

Simulation & Online & Offline (Reconstruction+Analysis)

- ❑ Be friendly to the user (not necessarily user friendly !)
 - ❑ PAW was a success
-

Toward AliRoot ...

The framework must be adapted to:

- The particularities of the experiment:
 - Large events: > 80 Mb
 - Many events: $> 2(+2)$ Pb/year
 - Huge data flow: > 1 Gb/s

Toward AliRoot ...

The framework must be adapted to:

- Any (most) user (about 900):
 - Rapidly usable (even if unstable)
 - Available on all major platforms
 - Low learning threshold

Looks like *a-posteriori* design criteria for Root

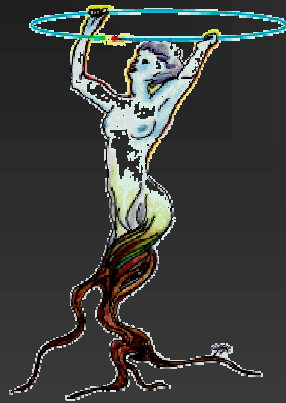


ALICE Off-line Project

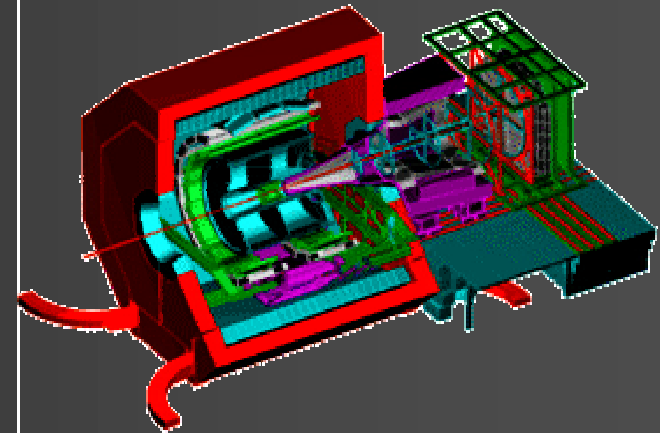
by Federico Carminati

ROOT

An Object-Oriented
Data Analysis Framework



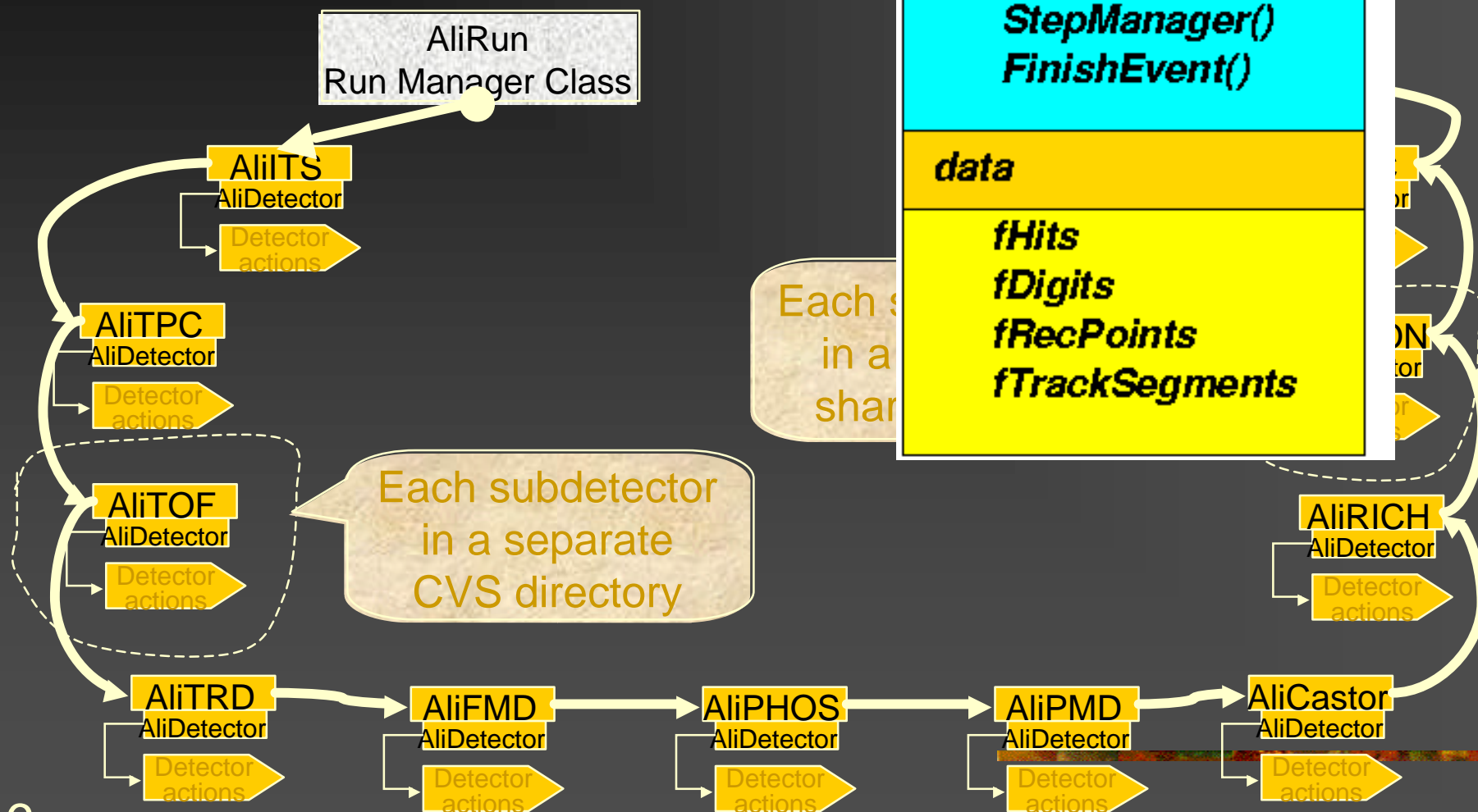
&



- All OO & C++
- The unique framework is **AliRoot** (same language for script and implementation)
- Fortran is tolerated (GEANT3.21 wrapped)
- Open source (user = potential developer)

An often disputed (at CERN) choice... but it works !

Organisation



Alice User

Root dialog



ALICE user

```
[schutz@subatech01 ~/work]$ alroot  
Constant Field Map1 created: map= 1, factor= 1.000000  
*****  
*                                     *  
*   W E L C O M E to R O O T         *  
*                                     *  
*   Version  2.23/11  14 January 2000 *  
*                                     *  
*   You are welcome to visit our Web site *  
*   http://root.cern.ch                 *  
*                                     *  
*****
```

```
CINT/ROOT C/C++ Interpreter version 5.14.25, Nov 25 1999  
Type ? for help. Commands must be C++ statements.  
Enclose multiple statements between { }.  
root [0] gAlice->Init("PHOSConfig.C");  
root [0] gAlice->Run(1,"PHOSConfig.C");
```

Talks to root
Through a
macro



Analyzer



Alice User



ALICE user

Talks to root
Through a
macro

Root dialog

```
void Config()
{
  TFile *rootfile = new TFile("alice.root","recreate"); // root file
  rootfile->SetCompressionLevel(2);
  new AliGeant3("C++ Interface to Geant3");           // transport model
  TGeant3 *geant3 = (TGeant3*)gMC;
  geant3->SetTRIG(1);           // Number of events to be processed
  geant3->SetSWIT(4,10);
  geant3->SetDEBU(0,0,1);
  geant3->SetLOSS(2);
  geant3->SetMULS(1);
  geant3->SetRAYL(1);
  Float_t cut   = 1.e-3; // 1MeV cut by default
  Float_t tofmax = 1.e10;
  geant3->SetCUTS(cut,cut, cut, cut, cut, cut, cut, cut, cut, cut, tofmax);
  AliGenCocktail *gener = new AliGenCocktail(); // Cocktail generator
  gener->SetPtRange(.02,10.00); // Transverse momentum range
  gener->SetPhiRange(180.,360.); // Azimuthal angle range
  gener->SetYRange(-0.25,0.25); // Pseudorapidity range
  gener->SetOrigin(0,0,0); // Vertex position
  gener->SetSigma(0,0,5.6); // Sigma in (X,Y,Z) (cm) on IP position
  gener->Init();
}
```


Alice User



ALICE user

Talks to root
Through a
macro

Root dialog

```
void Config()
{
  gAlice->SetField(0,0); //Specify maximum magnetic field in Tesla
  Int_t iMAG=1;
  Int_t iITS=0;
  Int_t iPHOS=1;
  //===== Alice BODY parameters
  AliBODY *BODY = new AliBODY("BODY","Alice envelop");
  if(iMAG) {
    //===== MAG parameters
    // --- Start with Magnet since detector layouts may be depending ---
    // --- on the selected Magnet dimensions ---
    AliMAG *MAG = new AliMAG("MAG","Magnet");
  }
  if(iPHOS) {
    //===== PHOS parameters
    AliPHOS *PHOS = new AliPHOSv0("PHOS","GPS2");
    Float_t x = (Float_t) thickness / 100. ;
    if (thickness == 0)
      x = 0.00001 ;
    PHOS->GetGeometry()-> SetLeadConverterThickness(x);
  }
}
```

Analyzer

Alice User



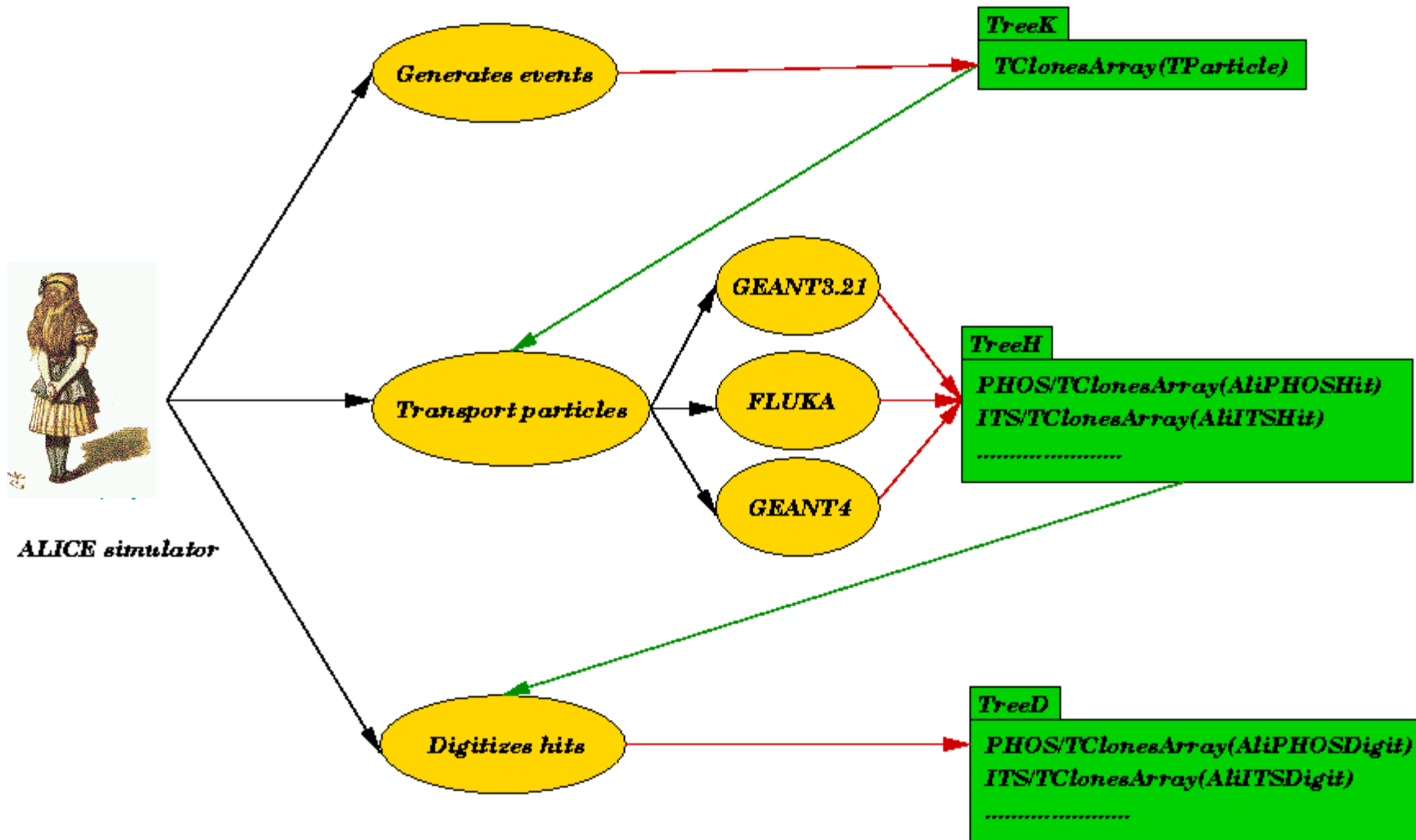
ALICE

Talks to
Through
macro

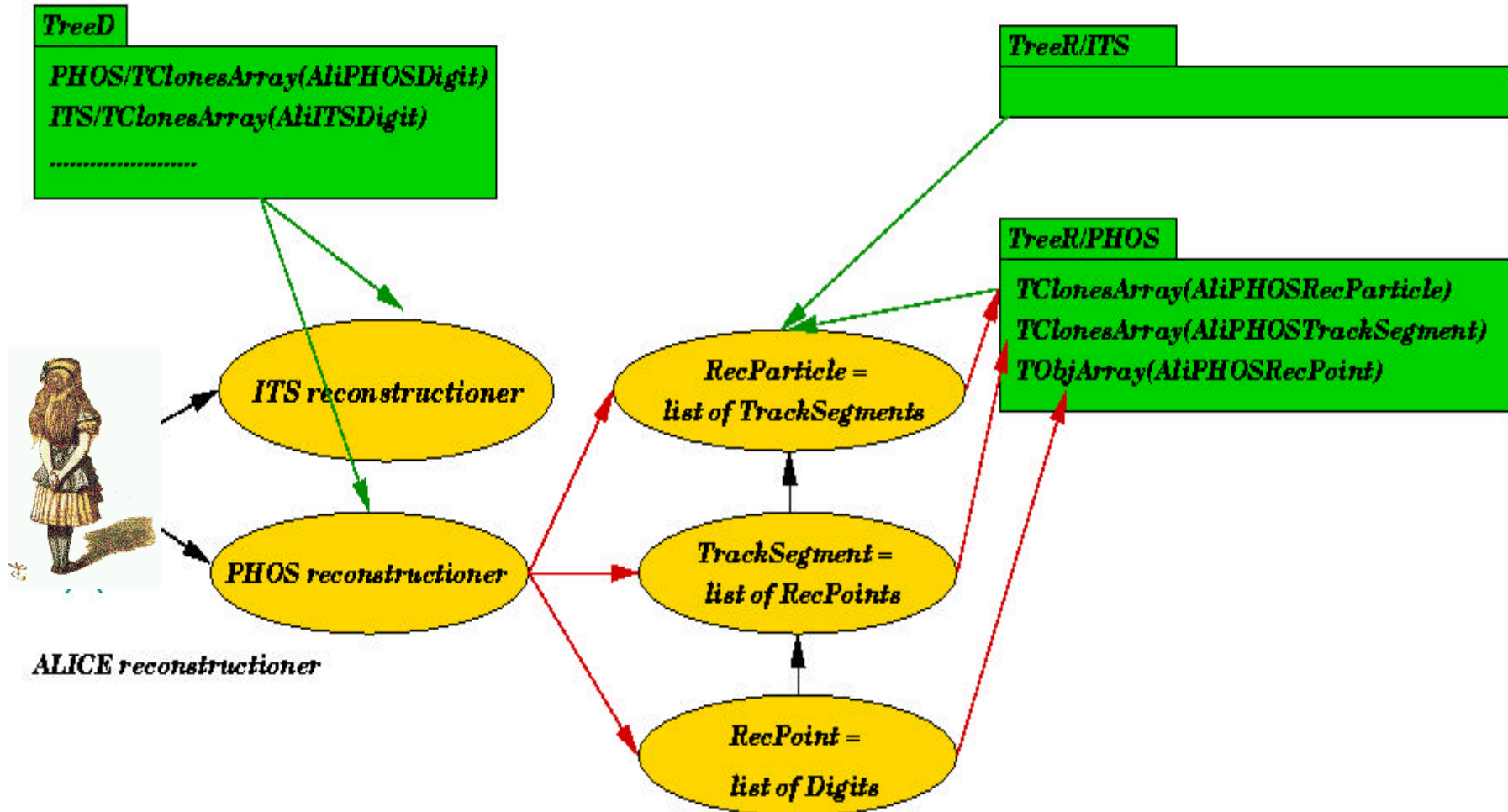
```
void Reconstruct()
{
  fRootFile = new TFile("MyFileName", "update") ;
  gAlice = (AliRun*) fRootFile->Get("gAlice");
  fPHOS = (AliPHOSv0 *)gAlice->GetDetector("PHOS") ;
  fGeom = AliPHOSGeometry::GetInstance( fPHOS->GetGeometry()->GetName(),
                                         fPHOS->GetGeometry()->GetTitle() )
  ;
  fObjGetter = AliPHOSIndexToObject::GetInstance(fPHOS) ;
  fClu = new AliPHOSClusterizerv1() ;
  fClu->SetEmcEnergyThreshold(0.030) ;
  fClu->SetEmcClusteringThreshold(1.0) ;
  fClu->SetCalibrationParameters(0., 0.00000001) ;
  fTrs = new AliPHOSTrackSegmentMakerv1() ;
  fTrs->UnsetUnfoldFlag() ;
  fPID = new AliPHOSPIDv1() ;
  fPID->SetShowerProfileCuts(0., 0., 0., 0.) ;
  fPID->SetDispersionCutOff(0.34) ;
  fRec = new AliPHOSReconstructioner(fClu, fTrs, fPID) ;
  gAlice->GetEvent(evt);
  gAlice->TreeD()->GetEvent(0) ;
  fPHOS->Reconstruction(fRec);
  fRootFile->Close() ;
}
```

```
void Analyze()
{
  TFile rootfile("junk.root") ;
  gAlice = (AliRun *)rootfile.Get("gAlice") ;
  AliPHOSv0 * phos = (AliPHOSv0 *)gAlice->GetDetector("PHOS") ;
  AliPHOSGeometry::GetInstance( phos->GetGeometry()->GetName(),
                                phos->GetGeometry()->GetTitle() ) ;
  AliPHOSIndexToObject::GetInstance(phos) ;
  Int_t evt = 123 ;
  TClonesArray * recparticleslist = phos->RecParticles(evt) ;
  TIter nextrecparticle(recparticleslist) ;
  AliPHOSRecParticle * recparticle ;
  AliPHOSIndexToObject * please = AliPHOSIndexToObject::GetInstance() ;
  while ( recparticle = (AliPHOSRecParticle *)nextrecparticle() ) {
    AliPHOSTrackSegment * tracksegment = recparticle->GetPHOSTrackSegment() ;
    tracksegment->Print() ;
    Int_t numberofprimaries = 0 ;
    Int_t * prim = recparticle->GetPrimaries(numberofprimaries) ;
    for (Int_t i = 0 ; i < numberofprimaries ; i++)
      please->GimePrimaryParticle( prim[i] )->Print() ;
  }
  rootfile.Close() ;
  gAlice = 0 ;
  phos = 0 ;
  recparticleslist = 0 ;
}
```

Simulator



Reconstructioner



Developers and users

- About 900 potential users
- About 50 developers scattered around the world
 - Steering group at CERN (5 persons)
 - 1-2 developer-librarian/detector
 - About 5 (regular+occasional) developers/detector

To stay within the frame(work)

- The offline board
 - The steering group
 - The librarians
 - Meets every week (micro), 6 weeks (macro)
- Users training
 - 4 one-week meetings/year
 - Users + developpers
 - Presentations + classes

...but not enough

ALICE defined Coding Conventions

- Strictly enforced (blame/shame)
- Automatic tool
 - Syntax
 - Naming
 - Style
- make check
- *.viol

Maintenance

- CERN based CVS Repository
 - Librarian have exclusive commit rights
 - Developpers are automatically informed by e-mail
 - HEAD must compile, not necessarely bug free
 - Every 2 weeks a release (3.02) bug free, usable by any user
-

Documentation

- Root builds html documentation (to be improved)
- Root does reverse engineering (uml look alike)
- Every detector responsible for its own documentation
- Collected on a single WWW page



<http://AliSoft.cern.ch/offline>

General Information

- News
- Meetings
- User Support
- User Environment
- Project Organisation
- Offline Policy
- Mailing Lists

AliRoot

- How to run
- Manual
- Releases
- Code
- Macros
- Code Development
- Installation

Activities

- Simulation
- Reconstruction
- Visualisation
- Mass Storage
- Detector Databases

Detectors

- CASTOR CPV
- FMD ITS
- MUON PHOS
- PMD RICH
- START TOF
- TPC TRD
- ZDC



ALICE Off-line Project

Introduction

Welcome to the home page of the ALICE Off-line Project. This page and the following contain the description of the features of ALICE Off-line environment.



The ALICE Off-line Project has started developing the current framework in 1998. The decision was taken at the time to build the simulation tool for the Technical Design Reports of the ALICE detector using the OO programming technique and C++ as an implementation language.

This led us to the choice of ROOT as framework and GEANT 3.21 as simulation code. A prototype was quickly built and put in production. The experience with this was positive, and in November 1998 the ALICE Off-line project adopted ROOT as the official framework of ALICE Off-line.

AliRoot is the name ALICE Off-line framework for simulation, reconstruction and analysis. It uses the **ROOT** system as a foundation on which the framework and all applications are built.

Except for large existing libraries, such as GEANT3.21 and Jetset, and some remaining legacy code, this framework is based on the Object Oriented programming paradigm, and it is written in C++.

by **Federico Carminati**

Documentation

- Root builds html documentation (to be improved)
- Root does reverse engineering (uml look alike)
- Every detector responsible for its own documentation
- Collected on a single WWW page

