A New Physics Analysis Framework for H1

Andreas Meyer
University of Hamburg
Andreas.Meyer@desy.de

EPS, Budapest, 14 July 2001

Motivation

- Reorganize central physics tools software
- Reunify cuts - establish common particle concept
- Make expert knowledge persistent, ie. reusable
- Fast event selection
  Quick navigation over increasing amount of data
- Integrated analysis environment:
  One tool for data storage and interactive analysis
- Common and unique code reference (C++)
- Code portability between analyses and common software
- OO-paradigms to organize software design and maintenance

Scope

- H1 is a running experiment
- Reconstruction/simulation code remains stable
- Scope: reorganize the quickly changing part -
  ie. physics analysis code and data
- Constraint: need to sustain quality & precision
- Development: core group effort
- Goal: interface between H1 data/software and user

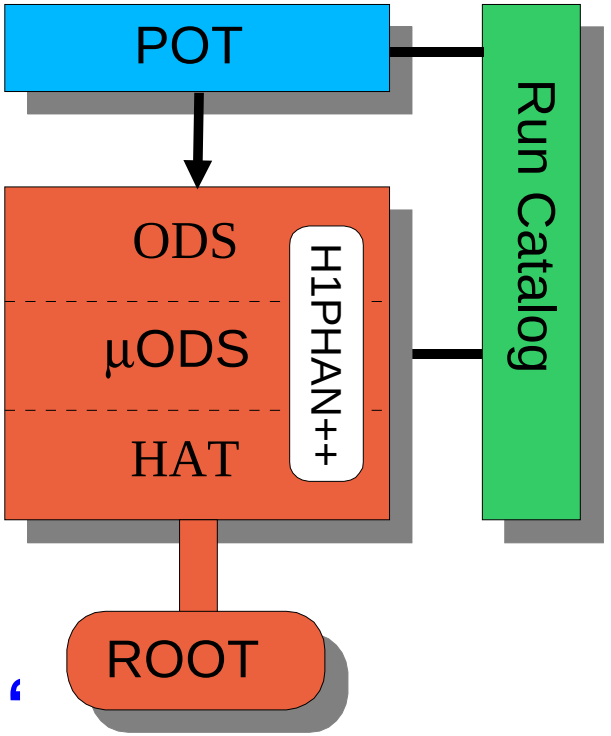
Nice and easy-to-use modern set of tools

Overview

- Data Storage Model
  - 3-layer file structure with persistently stored object relations: ODS, μODS, HAT
- Access to Data
  - H1Tree / RunCatalog
- Physics Algorithms
  - Coherent data, portable algorithms
- Event Display
  - Integrated with analysis environment

New Storage Model

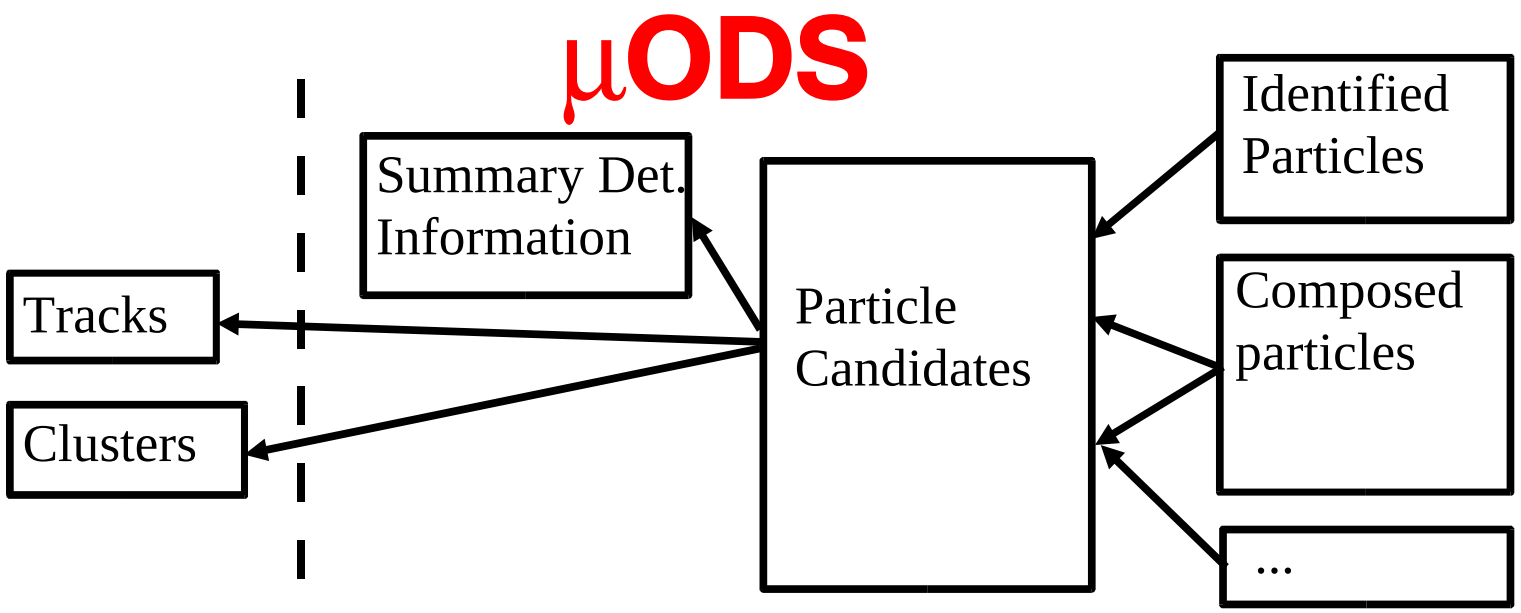
- Based on ROOT
- Three hierarchical layers
  - Reconstruction: 'ODS'
    (Object Data Storage, 13 kB/evt)
  - Particles: 'µODS' (1 kB/evt)
  - Event Tag: 'HAT' (0.4 kB/evt)
- Additional layer: 'User Tree'
  - RunCatalog: Retrieve data by run and event #



Common, unique environment for both H1 and user data

Three Layer Data Storage

- ODS: reconstruction level
  - Tracks, clusters, detector information
  - 1-1 correspondance with former DST
  - Backward compatibility:
    Existing analysis software remains functional
- μODS: particle level
  - 4-vectors of Id‘d particles
  - ”Intelligent” accessors
- HAT: event level
  - Concise event summary information
  - -> Fast event selection

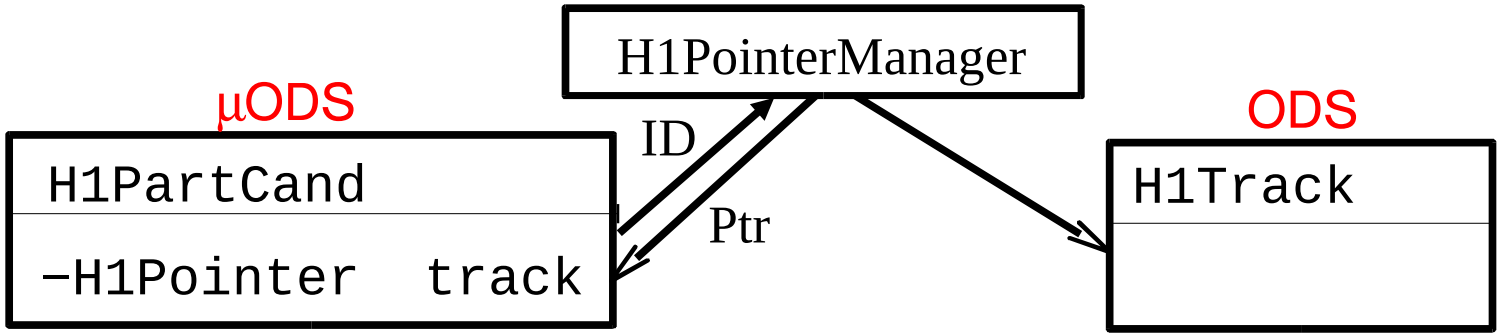


- Particle Candidates: particle kinematics
  Pointers to reco-level tracks and and clusters (ODS)
- Identified/Composed Particle Lists:
  Pointers to specific, selected particle candidates (extendable)
- In addition: Summary Detector Information:
  Fast information for specific tracks and clusters

Quick Navigation over Particles,
Direct Access to Tracks and Clusters

H1Pointer

- Persistent storage of class relations (also across different files, e.g. μODS->ODS)
- Direct access from particles (μODS) to complete reco-level information (ODS)



- Track and/or Cluster data automatically loaded when H1Pointer dereferenced

H1Pointer extends present functionality of RooT

H1Tree and RunCatalog

... fully encapsulate the data access

- Runcatalog (mySQL database)
  - contains relations between run/event # and files
- H1Tree:
  - Takes care of file handling (3 layers + user tree)
  - Event (pre-)selection on HAT -> event list
  - Direct access to selected events
  - Partial reading of data

Event delivery according to individual user selection

μODS/HAT Production Sequence

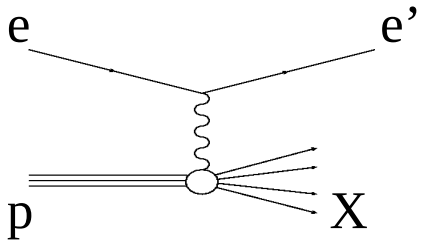
- Create Particle Candidates
  - Developed/provided centrally
- Identify/Find Specific/Composed Particles,
  - Developed/provided by individual analyses
    - e.g. with RooT-macros
  - Routinely run for μODS and HAT production
    - on all events, compiled C++/f77
- Physics code is modular
  - Implemented in (loosely coupled) 'Finders'-modules
- Keep physics code portable

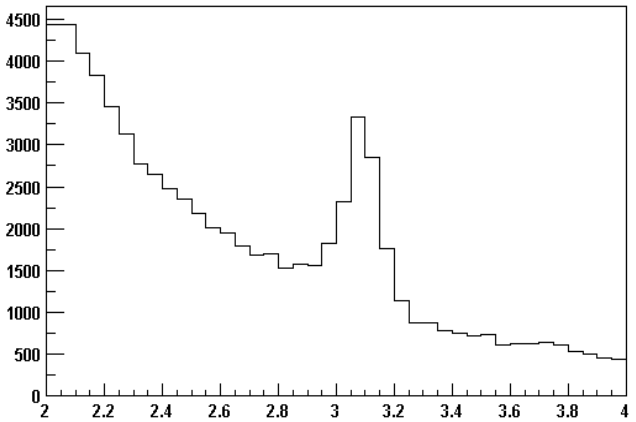
Goal: analysis code will be production code

Particles

Persistent storage of physics output:

- Scattered Electron (NC: ep->e‘X)
- Other electromagnetic particles
- Muons
- Selected Charged Particle Tracks
- Hadronic Final State Objects
- Jets (Kt, Jade)
- J/Ψ->ll, D*->Kππ
- ... continuously being extended





Results from expert analyses readily available to all H1

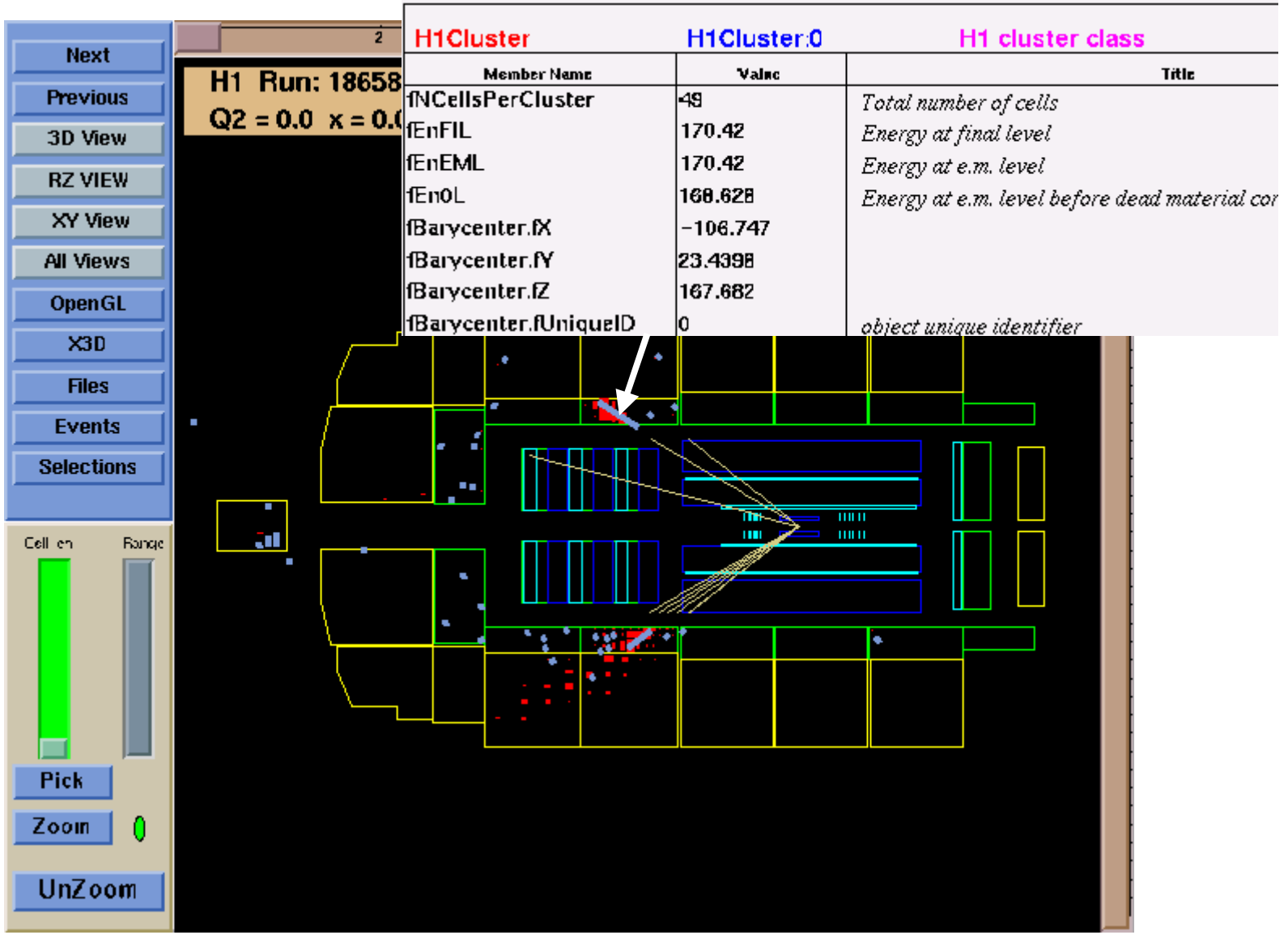
First Physics Analyses

- First complete production release (May 2001)
- First physics analyses in new framework are well underway
- Main concerns at present:
  - Add more analyses
  - Improve common particle concept
    Optimize cuts/calib. for full precision of all analyses
    (Track/cluster matching and calibration)
  - Improvement of speed/performance
    Optimize partial reading of data
    Straighten out implementation

Event Display

- Completely integrated in analysis framework
  - e.g. event selection on HAT
  - display events from within analysis code
- Modern GUI
- Pick and inspect
- Software from previous display largely re-used
- Can display data in RooT and BOS format
- Also accepts commands/macros of previous command-line based event display

An Event


Next
Previous
3D View
RZ VIEW
XY View
All Views
OpenGL
X3D
Files
Events
Selections
Cell en
Range
Pick
Zoom
UnZoom
H1 Run: 18658
Q2 = 0.0 x = 0.
H1Cluster
H1Cluster:0
H1 cluster class
Member Name
Value
Title
fNCellsPerCluster
49
Total number of cells
fEnFIL
170.42
Energy at final level
fEnEML
170.42
Energy at e.m. level
fEnOL
168.628
Energy at e.m. level before dead material cor
fBarycenter.fX
-106.747
fBarycenter.fY
23.4398
fBarycenter.fZ
167.682
fBarycenter.fUniqueID
0
object unique identifier

Summary

- ✔ A New Object-Oriented Physics Analysis Framework has been established:
  - ✔ based on RooT
  - ✔ persistent storage of expert analysis results
  - ✔ fast navigation of events, particles, reco-objects
  - ✔ coherent data formats, physics code portability
- ✔ A number of extensions to RooT:
  - ✔ H1Tree/RunCatalog: file handling / event delivery
  - ✔ H1Pointer: persistent relations across files
- ✔ A fully integrated event display

Conclusions

- ✔ Introduction of a New Analysis Framework in a Running Experiment Has Been Proven Successful
- ✔ Present and Future:
  - ✔ Continuously Optimize Physics Analysis Capabilities:
    - ✔ Integrate New Analyses
    - ✔ Establish Common Particle Concept
    - ✔ Improve Speed Performance

Physics Analyses Will Greatly Profit

From New Superior Software Environment