Event Processing

Stephen J. Gowdy
SLAC

Andrea Dell'Acqua
CERN

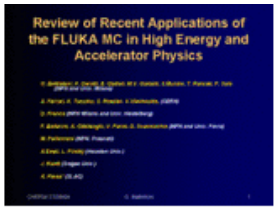
Computing in High Energy Physics 2004
Interlaken
1st October 2004

Overview

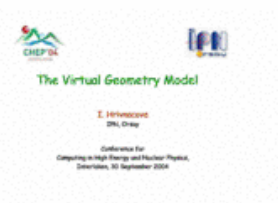
- Contributions
- Application Areas
  - Simulation
  - Trigger Algorithms
  - Data Models
  - Reconstruction
  - Graphics
  - Analysis Tools
- Observations, thoughts, questions

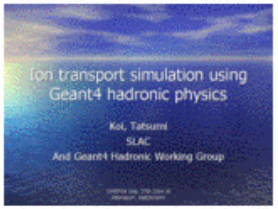
Contributions

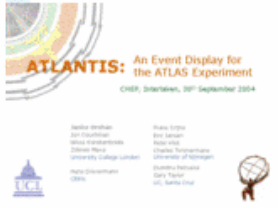
- 48 Parallel Talks
  - No time to talk about everything said in 20 mins, which pick a few samples in each area
- 34 Posters
  - Not going to talk about posters

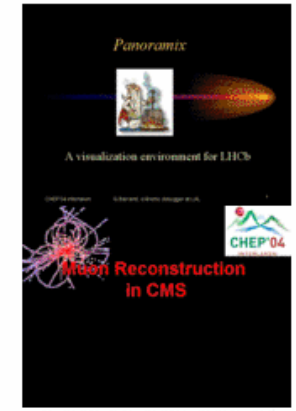


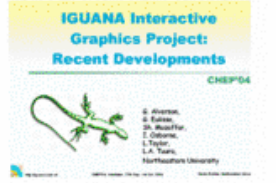
The GeoModel Toolkit for Detector Description

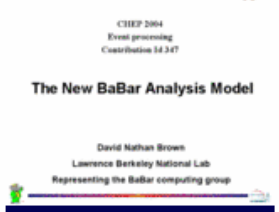


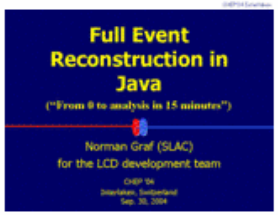


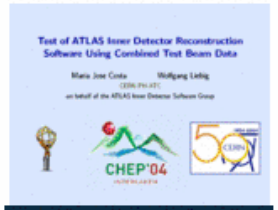


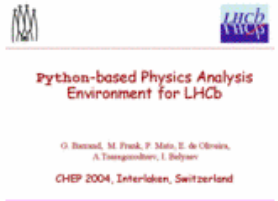


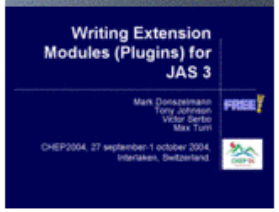


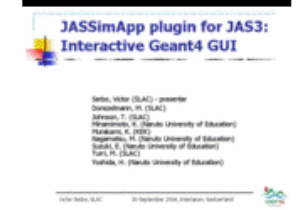


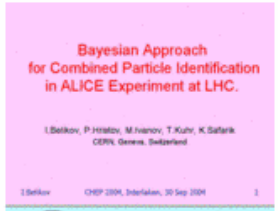


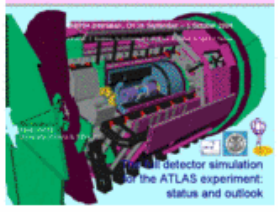


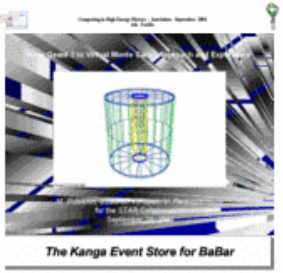


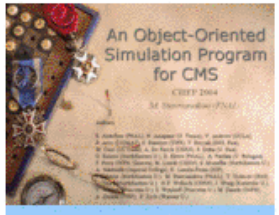






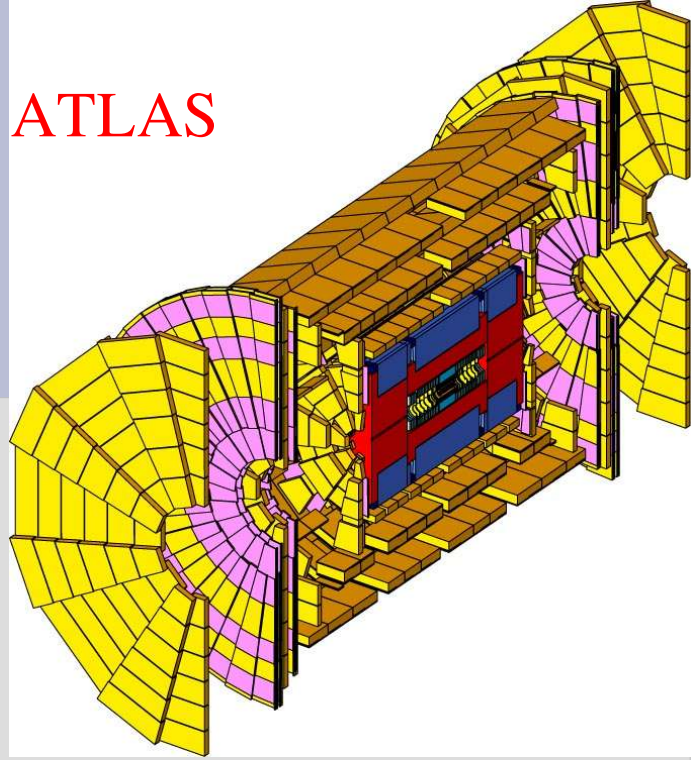






Simulation

- Switching from GEANT3 to GEANT4
  - Both CMS and ATLAS have now switched
  - ALICE and STAR also considering FLUKA
    - Source code to be available “soon”
    - Comparisons to data show similar agreements as GEANT4
- Fast simulations a must
  - For CMS reduce minutes to < 1s



Sliced view of CMS barrel

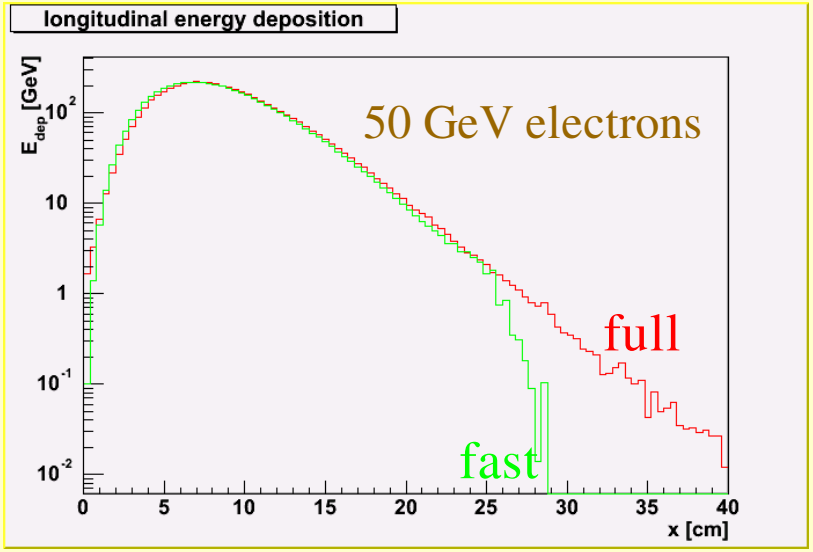
Parameterized Simulations

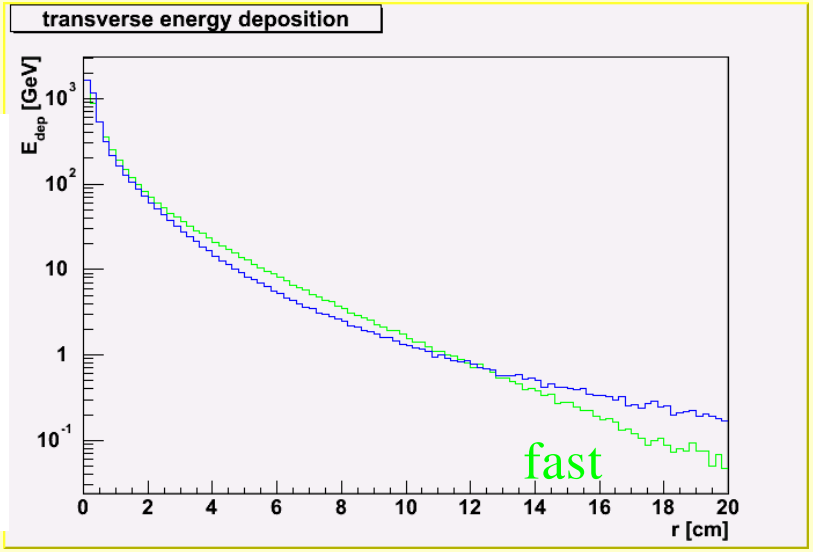
G4FLASH

Implementation of fast EM shower simulation in Geant4/OSCAR, using GFLASH parameterized showers (spot density)

- tuning in progress

Geant4 6.2



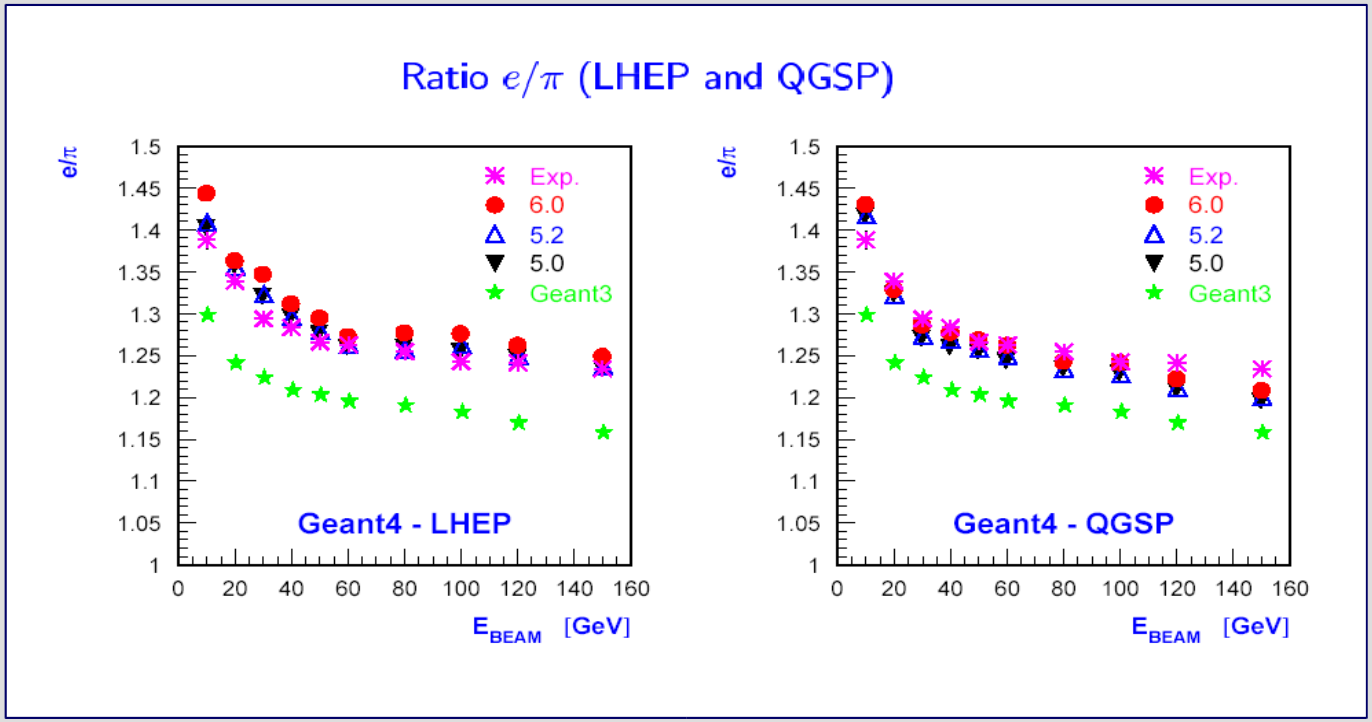


Timing studies

Electron energy	*Time/event full simulation*	*Time/event fast simulation*
1 GeV	0.8 s	0.5 s
10 GeV	1.9 s	0.6 s
100 GeV	16 s	0.7 s
300 GeV	57 s	1.0 s

Simulation (Cont.)

- A lot of work done on data validation
  - EM is “now at percent level, working on per mil”
- Most data/GEANT3/GEANT4 comparisons favour G4
  - but not all



ATLAS Calorimeter

Acceptance suite for Standard EM package

- To insure stability of results with time
- To control performance
- Is based on extended electromagnetic examples
- G4EmCalculator is an interface to dE/dx and cross sections
- Control on summary numbers:
  - Average energy deposition
  - Shower shape
  - Scattering angles
- Tests on cross sections and dE/dx
- Comparison of histograms (in project)

Συνεργεια

Summary

- Synergia is a distributable beam dynamics framework
  - incorporates and extends existing codes
  - provides job management system
  - documentation & test suite
- Already used for beam studies
- Will be extended to include more physics
  - requires some framework development

Trigger Algorithms

- Seeing some commonality in algorithms and software between high level trigger and offline reconstruction
- Work going on in optimising speed various ways
  - Fast algorithms
  - Limiting to regions of interest
  - And ...

Run the fastest and most likely to reject algorithm first

Optimisation of the evaluation sequence

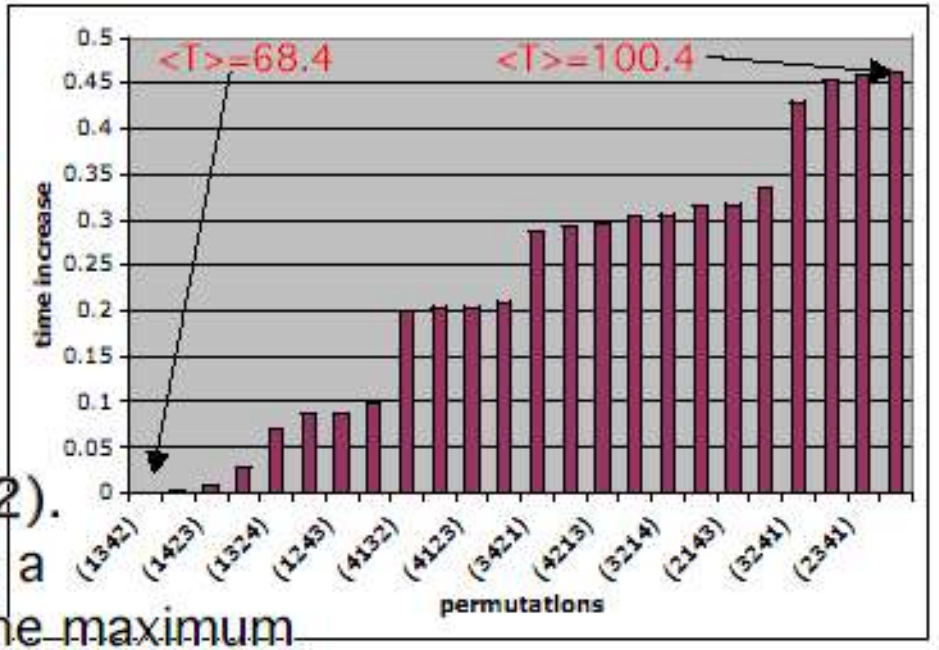
UCL

- Evaluation of all possible combination grows as n!
- Define an order between two sub trigger as:

$$\|kl\| < \|lk\| \equiv pa_k t_k + (1 - pa_k) pa_l (tr_k + ta_l) < pa_l t_l + (1 - pa_l) pa_k (tr_l + ta_k)$$

- Sort the sequence $\{k_1 \ldots k_n\}$ according to the defined order.

bit	p_i	t_a	t_r
1	0.5	60	12
2	0.3	60	50
3	0.1	20	10
4	0.6	90	40



- Best combination is (1342).

Other combination result in a time increase of 50% for the maximum

e/γ Selection – Triggering for Higgs Bosons

Results from the studies on single & double electron triggers were used to evaluate the HLT efficiency for selecting a fully simulated Higgs candidate in the range:

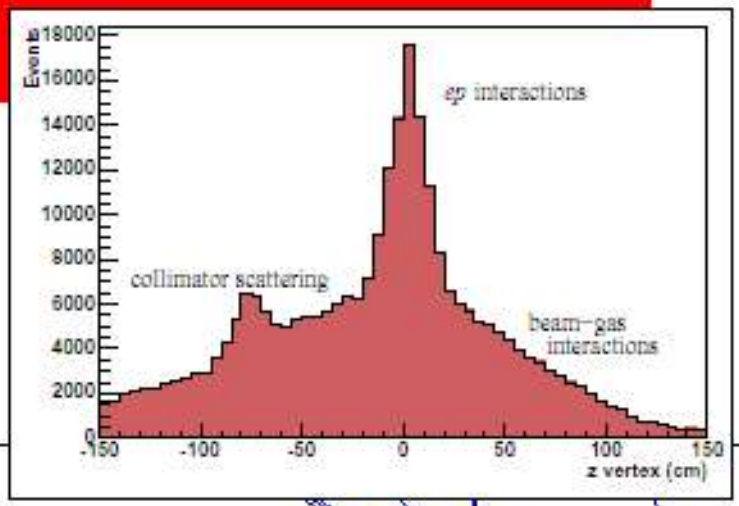
$130 < m_H < 180$ (GeV)

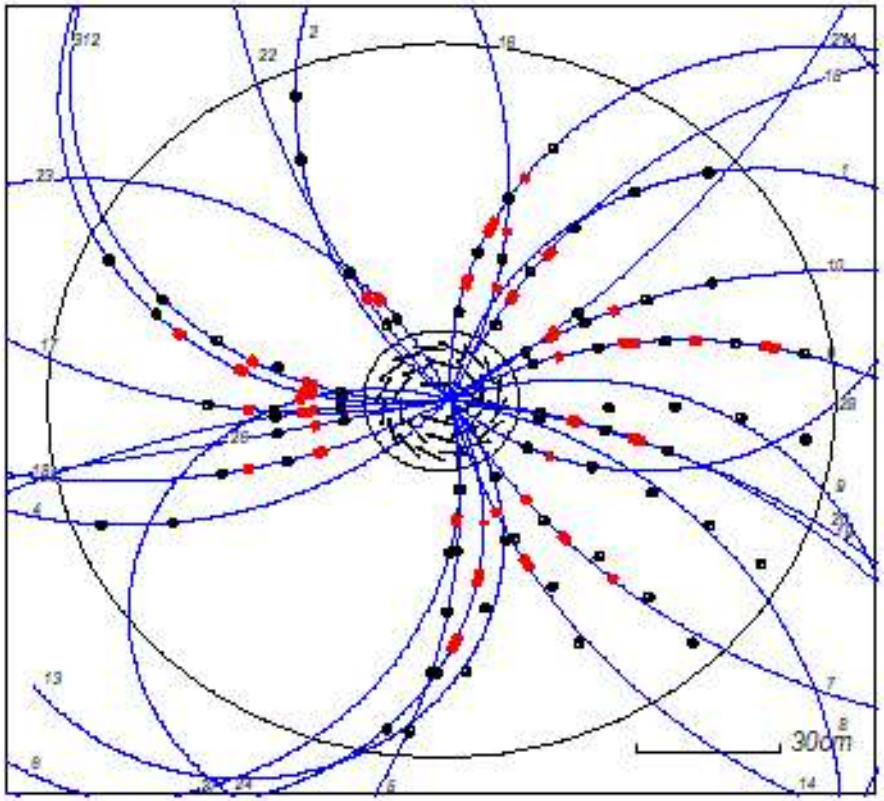
For different combinations of trigger items, we checked the Higgs selection efficiency with respect to an initial sample*

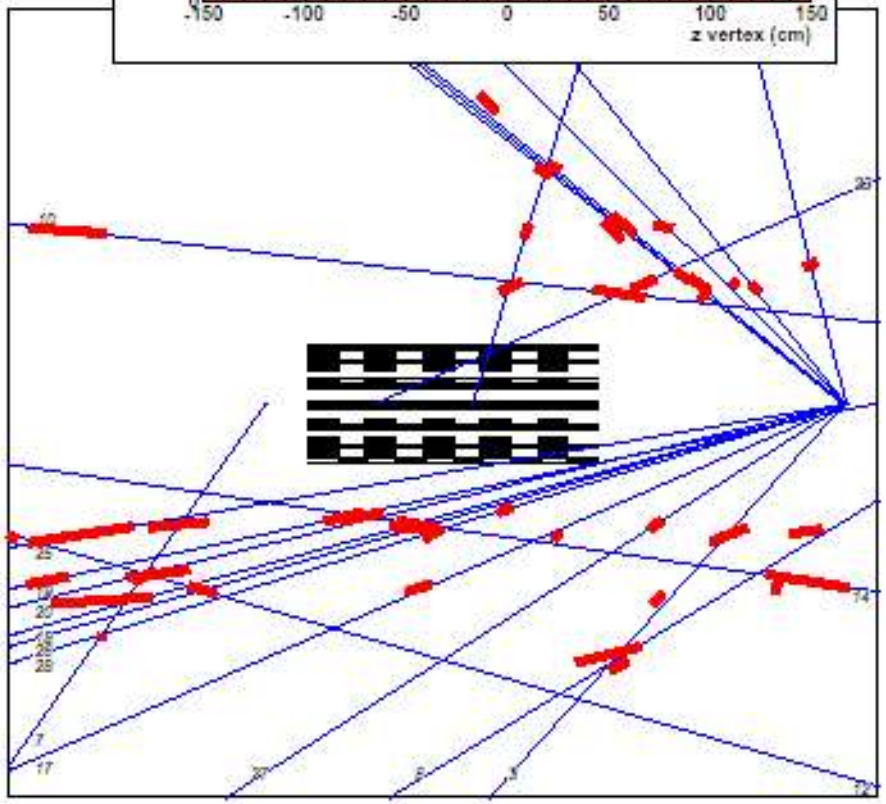
Trigger	Luminosity	H ?4e (130GeV)
e25i	*Low(startup)*	96.5 ± 0.2
2e15i	*low*	95.8 ± 0.2
e25i or 2e15i	*low*	96.7 ± 0.2
e30i	*design*	96.0 ± 0.4
2e20i	*design*	94.5 ± 0.4
e30i or 2e20i	*design*	95.5 ± 0.3

Event Topology – Real Data (contd)

- Beamgas scattering from -80cm collimator.



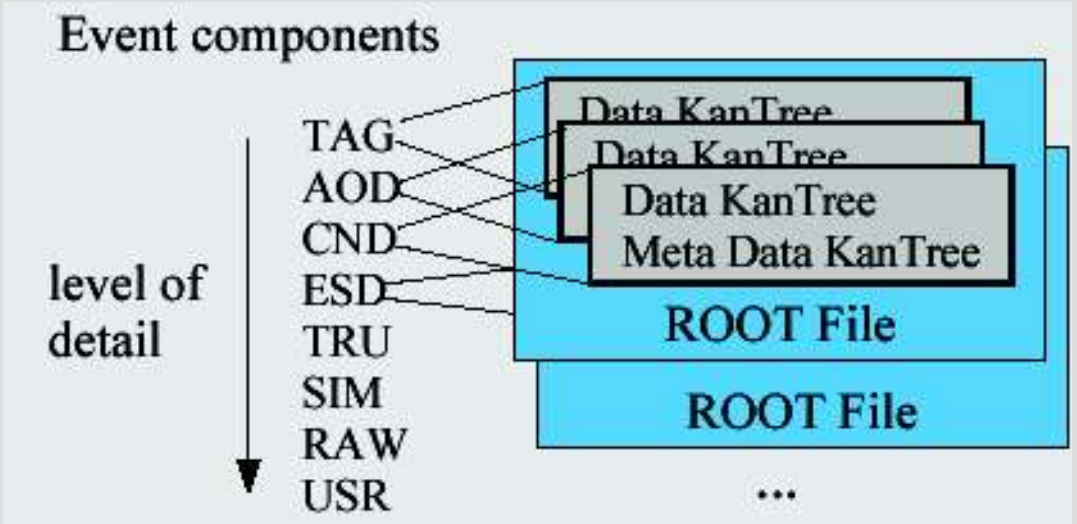




Data Models

- Heard about two data models
  - BaBar's Computing Model 2
  - ATLAS Event Data Model

172: ... Kanga Event Store for BaBar
Component layout in files



Distinct objects in different levels

Track

One of the most important elements in the ATLAS EDM is the common **Track** (an **ESD** level object).

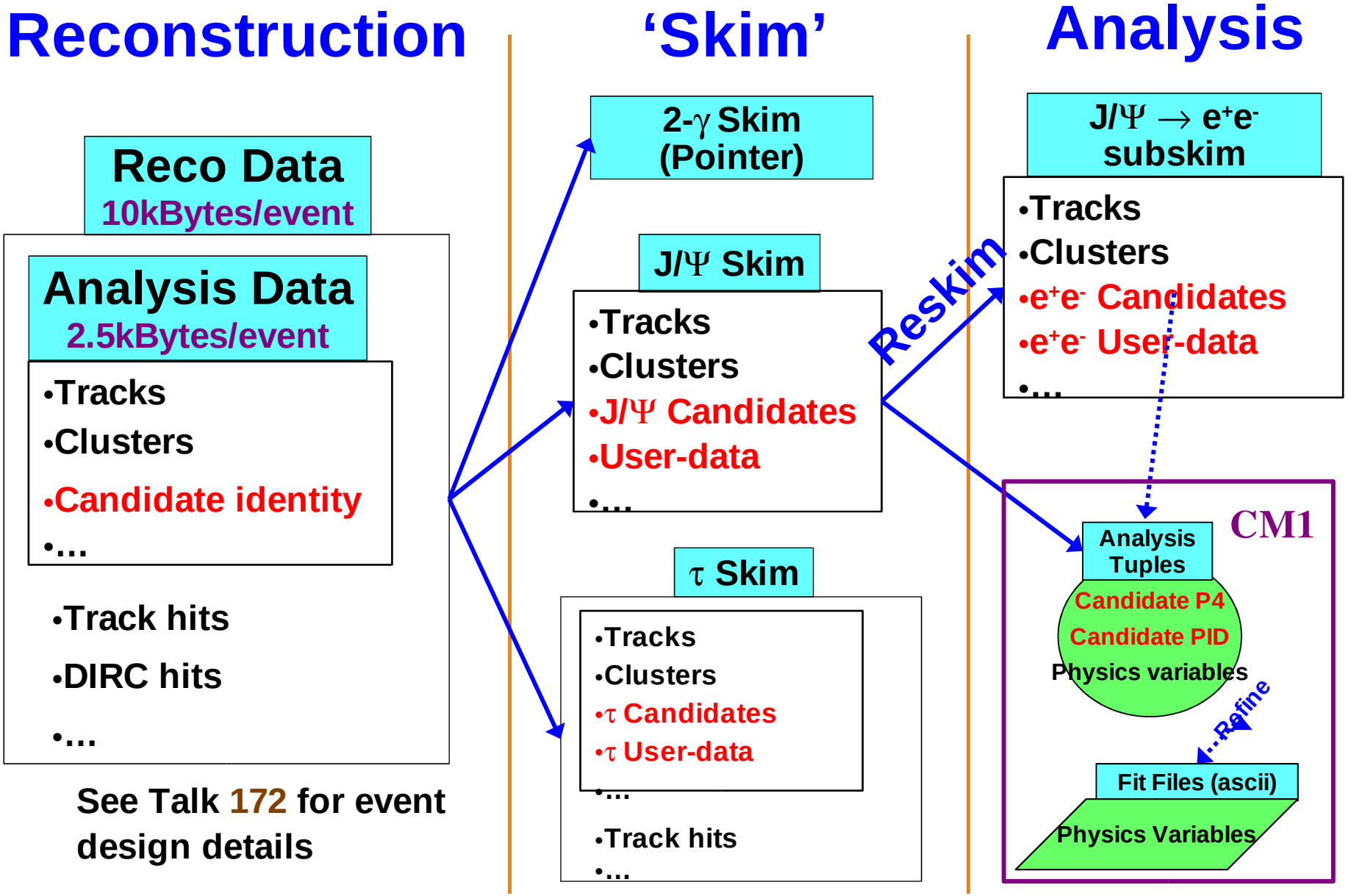
- It must work in a wide range of applications, from
  - online (where speed is important)
  - alignment studies (which need detailed information)
  - reconstruction
- Tracks at **ESD** level consist of fitted measurements on multiple surfaces
- It is the *output* from the fitters, and is the *input* to the combined reconstruction.
- All reconstruction packages use the same track class.

For **AOD**, something more lightweight is needed: **TrackParticles** are created from Tracks:

- Contain summary information about parent track (number of hits on track etc)
- Are physics analysis objects, with 4-momenta (the class inherits from I4Momentum – in general AOD objects inherit from I4mom, and IParticle etc.)
- Can be used for vertex finding, but not re-fitting etc.

Different levels are the same object

CM2 Analysis Model

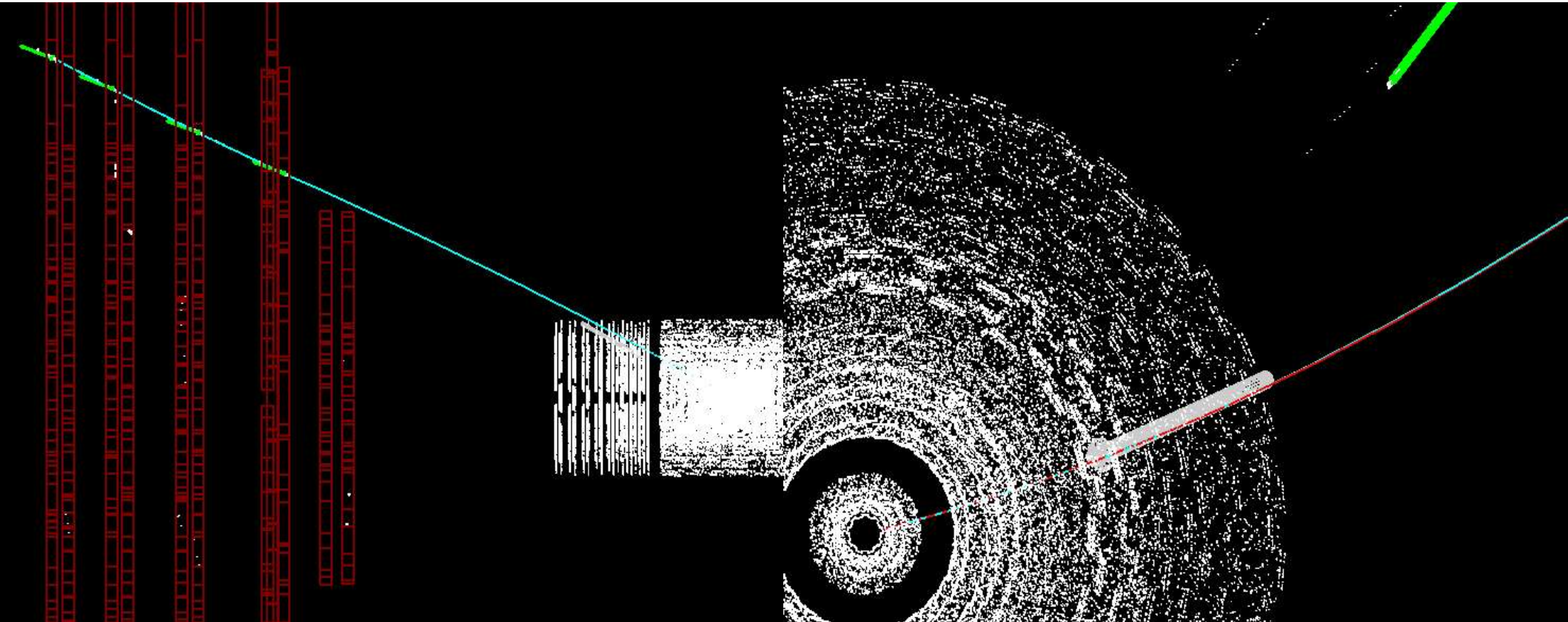


Reconstruction

- Various track finding algorithms reported
  - Some optimised for speed
  - Others to deal with high multiplicity environments
- Much discussion also on the Vertex finding and fitting algorithms
- 405: Event Reconstruction in Java
  - Used for North American work on International Linear Collider currently
  - Believed to be only 20-30% slower than C++

227: Muon Reconstruction Software in CMS

Find potential muons in muon system
Extrapolate back to outside of inner tracker and interaction region
Open window for track reconstruction based on Level2 track



229: Fast Reconstruction of tracks ... CBM experiment

Average of 4.8ms on CPU, w/ 8 50MHz FPGA 15 s!

Cellular Automaton Method

1. NIM A329 (1993) 262
2. NIM A387 (1997) 433
3. NIM A489 (2002) 389
4. NIM A490 (2002) 546

Define :

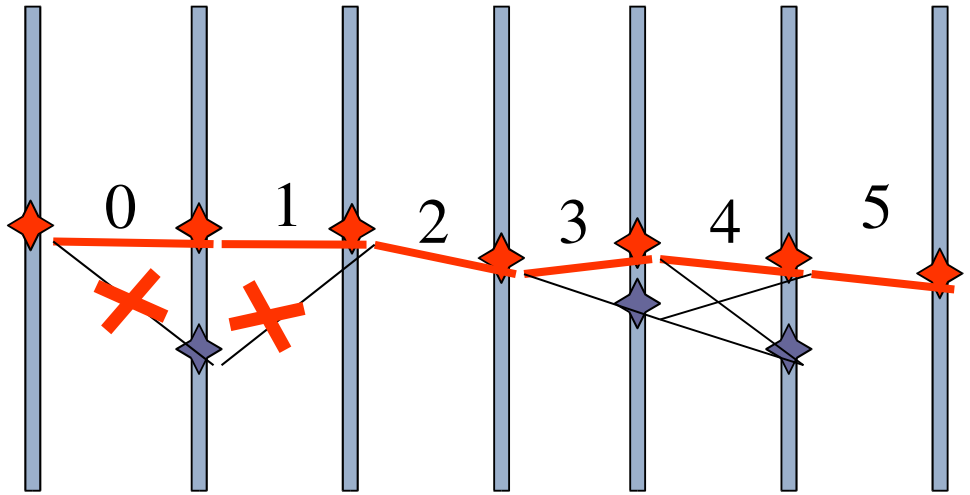
- CELLS -> TRACKLETS
- NEIGHBORS -> TRACK MODEL
- RULES -> BEST TRACK CANDIDATE
- EVOLUTION -> CONSECUTIVE OR PARALLEL

Being essentially **local and parallel** cellular automata **avoid exhaustive combinatorial searches**, even when implemented on conventional computers.

Since cellular automata operate with **highly structured information** (for instance sets of tracklets connecting space points), **the amount of data** to be processed in the course of the track search **is significantly reduced**.

Further **reduction of information** to be processed is achieved by **smart definition of neighborhood**.

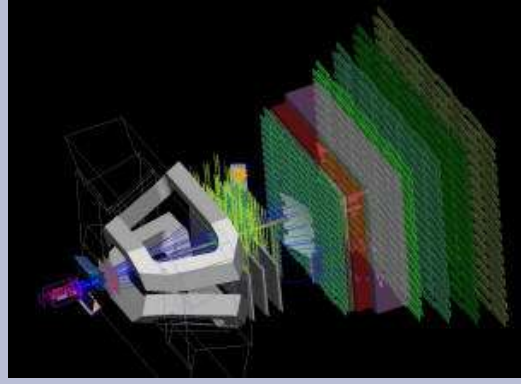
Usually cellular automata employ **a very simple track model** which leads to **utmost computational simplicity** and a **fast algorithm**.



Create tracklets

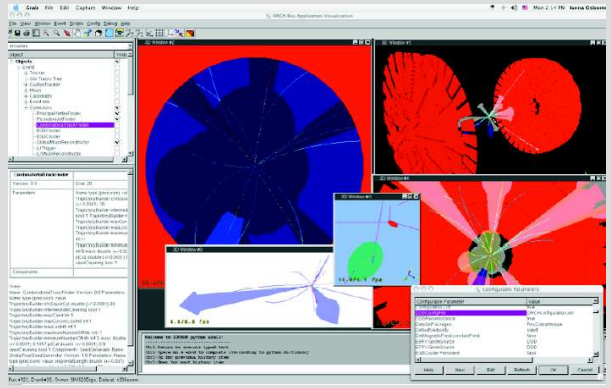
Collect tracks

Geometry

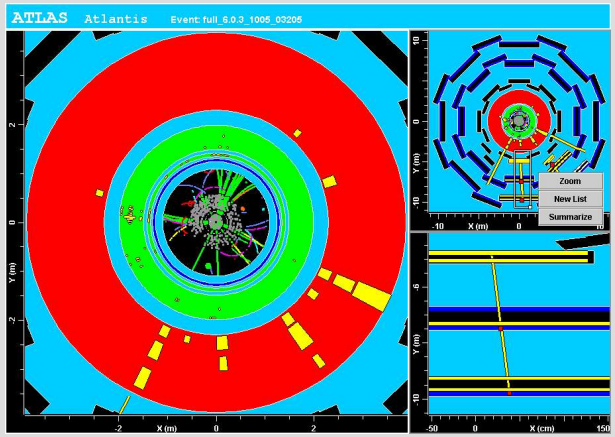


135: Panoramix ...

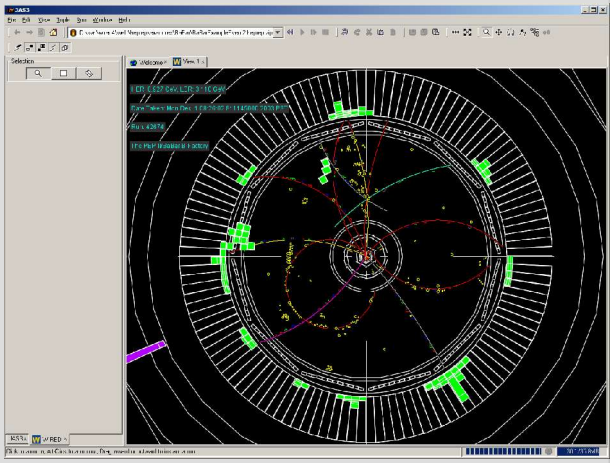
- Two geometry packages and four visualisation suites
- One (mostly 2D), three 3D



57: IGUANA ...



168: The Atlantis ...



395: Wired 4 ...

Conclusions

- The VGM introduces a general approach for conversion of geometries between specific geometry models
  - Geant4, Root TGeo, XML (AGDD, GDML)
  - This gives a possibility for a user of one specific package to use the tools supported by other packages:
    - Root TGeo => Virtual MC
    - XML (AGDD, GDML) => GraXML
- It also allows the user to define geometry independently from a specific geometry model
  - However this was not the main goal of the tool
- Available from
  - http://ivana.home.cern.ch/ivana/VGM.html

Analysis Tools

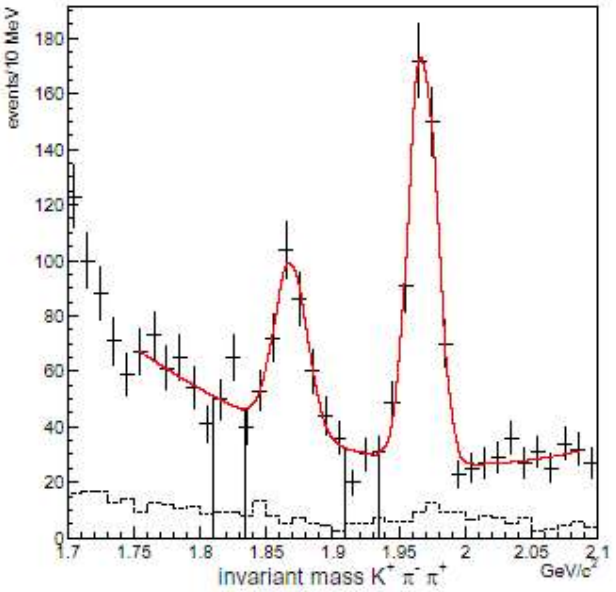
- 379: Python-based ... analysis ... for LHCb
  - No need for compilation
  - Ideal is one page of code for analysis
    - An example in the talk
- 347: ... BaBar Analysis Computing Model
  - Attempt to put all information in eventstore
    - Except the final fitting information
  - Users embracing new system

Evolved program shows where to look

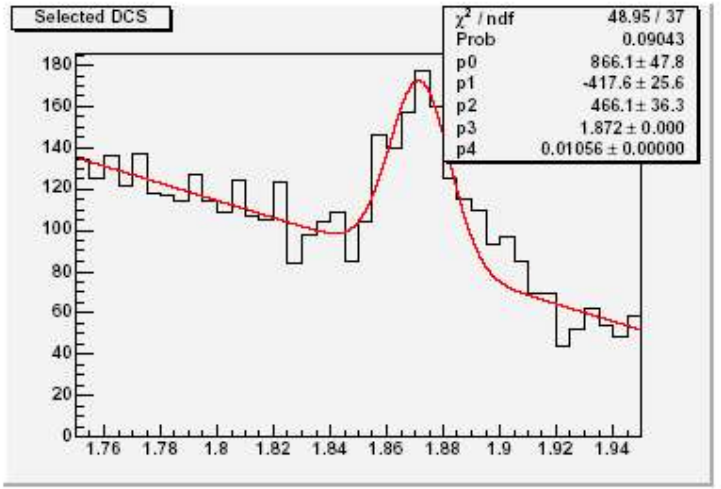
Comparison with Cut Method

How does this compare with our normal method?

- From hep-ex/0407014, measured BR of $D^+ \to K^+\pi^+\pi^-$
- Not a direct comparison, not optimized on $S/\sqrt{S+B}$



- Similar signal to noise
- Cuts: Yield $= 189 \pm 24$ events
- GP: Yield $= 466 \pm 36$ events



D_s^+ also shown

LHC Experiments

- Many experiments making huge progress
  - Now have foundations for first system
- CMS & ATLAS produced O(10M) events
  - O(day) worth of data but a big step
- Valuable experience from test beam(s)
    - (Only heard about ATLAS combined test beam)
  - Validating/tuning Monte Carlo
  - Detector and software experts find working relationships
  - Real data never quite the same as MC

Software Support

- Seems to be a growing “problem”
  - Upcoming experiments recognise the need
  - Ongoing experiments see loss of experts
    - Sited as one reason for FORTRAN -> C++ change
- Many reports of being memory leak free
  - Are we getting better at this?
- A couple mentions of issue of use vrs development
  - Want something stable but also need to deploy required features
  - Underscores need for at-scale test environments

Commercial/Closed Software

- Almost completely gone?
  - Did we hear the last gasp or are we missing something?
- Why?
  - Limits ability to respond to technology changes
  - At the whims of “vendor”
    - Could decide to change fee structure or no longer support product

Platforms

- Much of the software only works on Linux
  - Will this cause problems in the future?
- Alternatives?
  - Solaris
    - Has been around a while and still seen in “server” environment
  - Mac OS X
    - Growing popularity
      - Noticed on LCG poster they are now supporting this
    - Based on BSD and certainly different that Linux
    - Compiler is however the same
      - Well, could use Intel C++ on Linux

"Never write the same thing twice."

Collaboration on Software?

- Not much evidence for sharing software
  - Some good counter examples are ROOT, CLHEP and GEANT4, all "external" to experimental software itself though, not applications
- Most bad examples are copy and branch
  - eg EvtGen and BaBar Framework
- Can we do better?
- Should we?

OO Battle?

- Only legacy FORTRAN discussed
  - Still some being replaced, majority remaining seems to be physics generators
- Mostly C++
  - Some mention of going back to c?
- Java (still) mostly confined to graphics and event viewing
  - Places where performance isn't limiting factor
- Python becoming more popular
  - One big use is for quick “prototyping”

Conclusion

- Much progress in deploying initial LHC software
- Many “lessons learned” for everyone to share (or are we doomed to repeat them?)
- Very broad range of contributions
  - Trigger through analysis software
  - Only touched on a few talks, the rest are of course available on the web
- Validation seemed to be a recurring theme