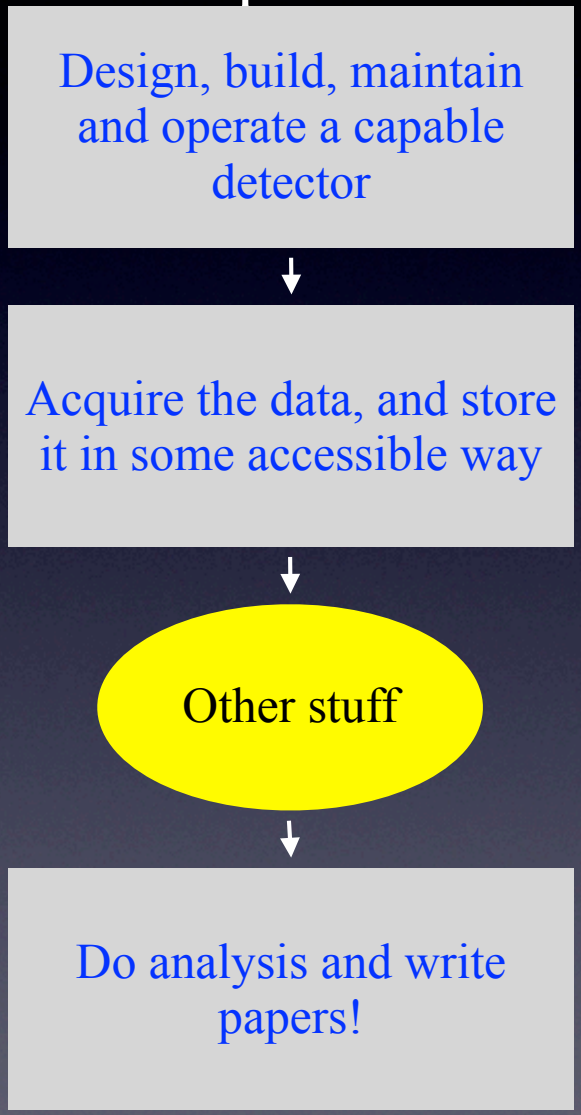
# DØ Algorithms

Erich W. Varnes
*University of Arizona*

DØ Collaboration Meeting
June 10, 2014

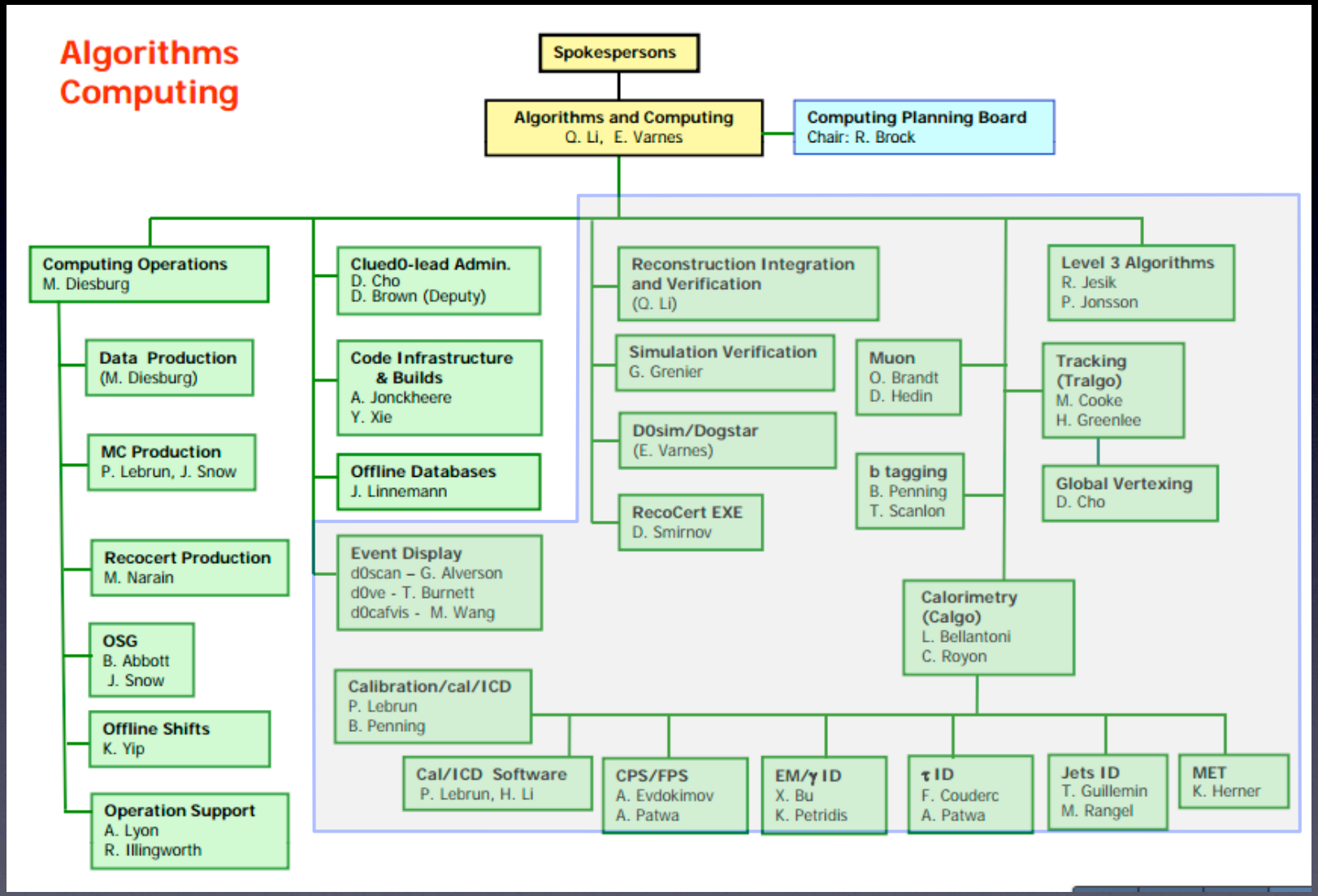
# DØ's Core Algorithms

- (Algorithm-centric view of ) Steps needed for a successful HEP experiment:




- All the "other stuff" is what we call the core algorithms
- Includes
  - Reconstruction
  - DØ-specific parts of MC simulation
  - Object ID tools

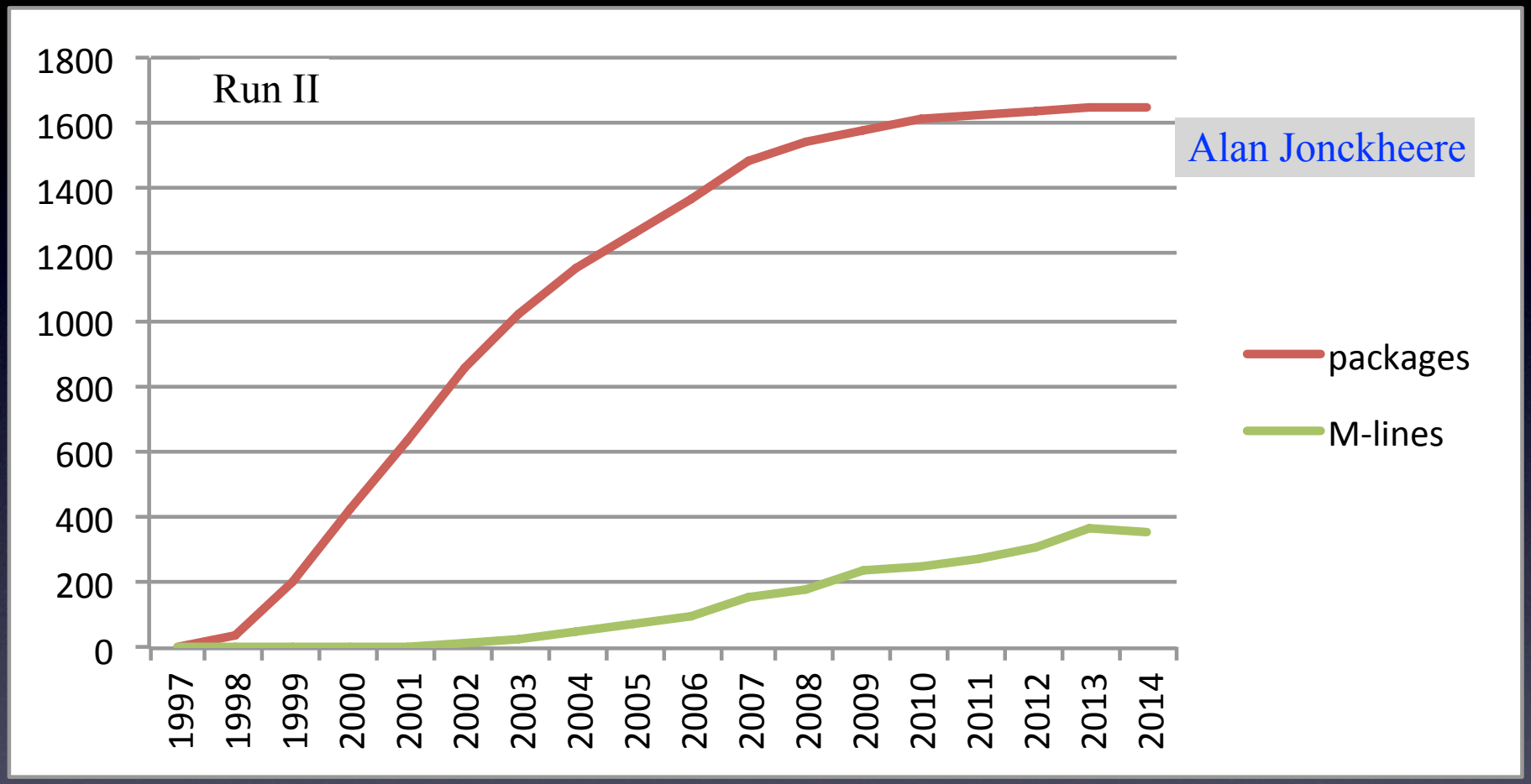
# DØ's Core Algorithms




ca. 2010

# DØ's Core Algorithms

- These algorithms form the majority of the code base in d0cvs:



# Caveats

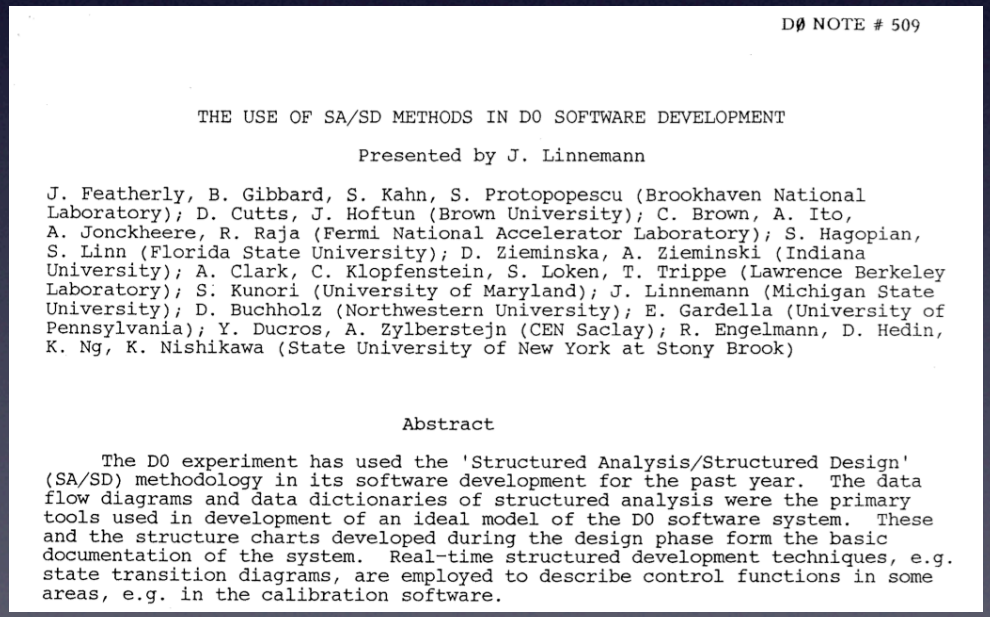
- Way too much material to cover in 20 minutes
- I will mention some people who were critical to the development of our algorithms
  - which inevitably means I will leave out others who also deserve to be mentioned
- Discussion will be colored by my own perspective

Part of that perspective was formed then → (back when the Cleveland Browns were competitive)

# The Beginning: Run I

- 4π collider detectors were relatively new at the time
  - much of DØ had come from fixed-target background
- Soon realized that DØ reconstruction was a far more complex task
  - effort to get d0reco going led by Serban Protopopescu

- Structured Analysis/ Structured Design method was seen as key in organizing code
  - provided needed initial push, but not followed throughout development



DØ NOTE # 509

THE USE OF SA/SD METHODS IN D0 SOFTWARE DEVELOPMENT

Presented by J. Linnemann

J. Featherly, B. Gibbard, S. Kahn, S. Protopopescu (Brookhaven National Laboratory); D. Cutts, J. Hoftun (Brown University); C. Brown, A. Ito, A. Jonckheere, R. Raja (Fermi National Accelerator Laboratory); S. Hagopian, S. Linn (Florida State University); D. Zieminska, A. Zieminski (Indiana University); A. Clark, C. Klopfenstein, S. Loken, T. Trippe (Lawrence Berkeley Laboratory); S. Kunori (University of Maryland); J. Linnemann (Michigan State University); D. Buchholz (Northwestern University); E. Gardella (University of Pennsylvania); Y. Ducros, A. Zylberstejn (CEN Saclay); R. Engelmann, D. Hedin, K. Ng, K. Nishikawa (State University of New York at Stony Brook)

Abstract

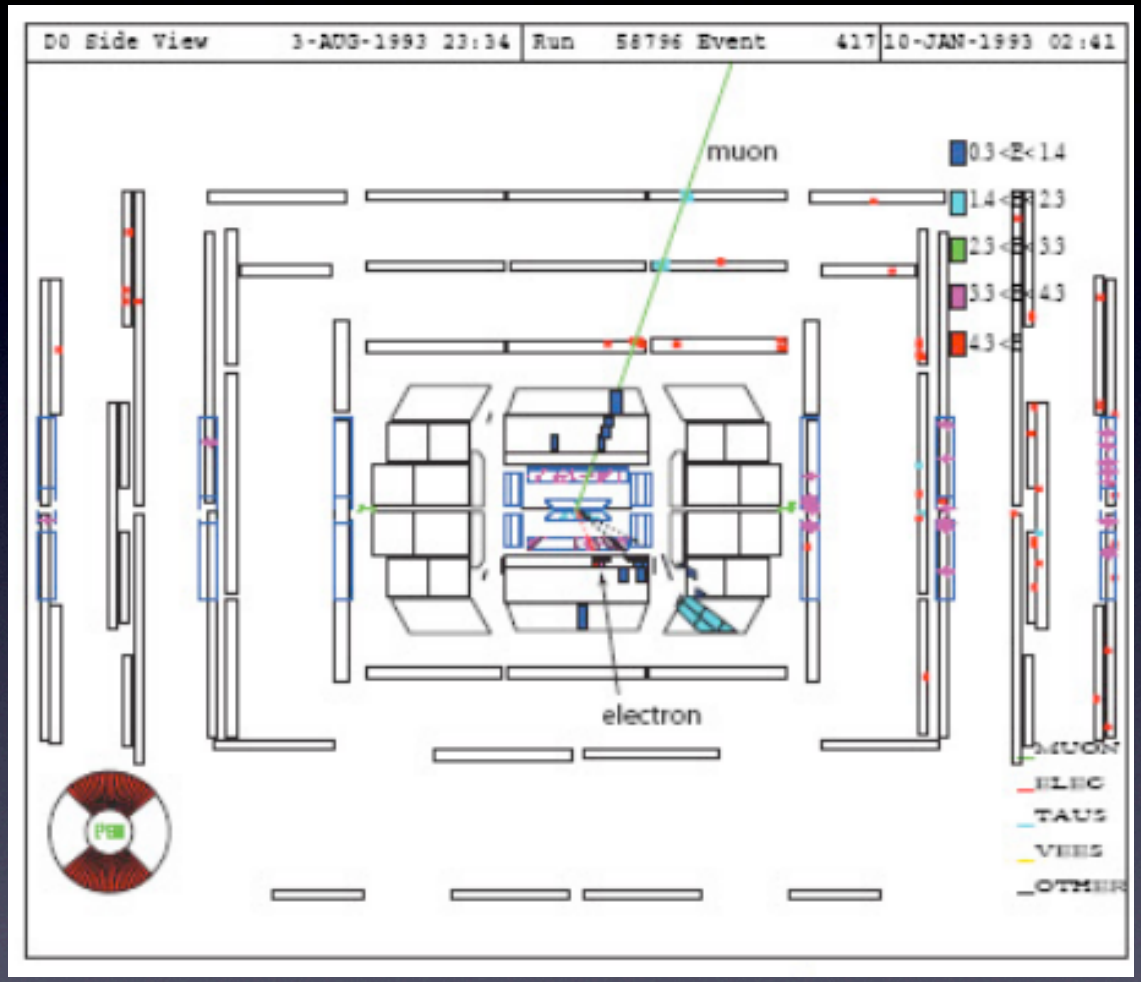
The D0 experiment has used the 'Structured Analysis/Structured Design' (SA/SD) methodology in its software development for the past year. The data flow diagrams and data dictionaries of structured analysis were the primary tools used in development of an ideal model of the D0 software system. These and the structure charts developed during the design phase form the basic documentation of the system. Real-time structured development techniques, e.g. state transition diagrams, are employed to describe control functions in some areas, e.g. in the calibration software.

# Developing DØreco

- Software throughout HEP was written in FORTRAN
  - nice for computation, but little in the way of memory management
- Smaller experiments had gotten by with COMMON blocks
  - chunks of memory that all routines could read/write
  - problem: if routine A writes something where routine B expects something else...
- Solution adopted was the ZEBRA memory management system
  - information organized into various “banks” for electrons, muons, jets, etc.
  - somewhat similar to C++ classes (or at least C structs)

# To Make a Long Story Short

- It worked!



# From FORTRAN to C++

- In the period between Runs I and II the field of HEP underwent a transition from VAX/FORTRAN to (something)/ C++
  - yes, we discovered the top quark without ever using a class*
  - eventually (something) became essentially Linux
- DØ was at the front line of this transition
  - programming concepts are dramatically different
  - C++ has many more features
    - but which ones should be used when?
    - how should one trade off between flexibility and efficiency?

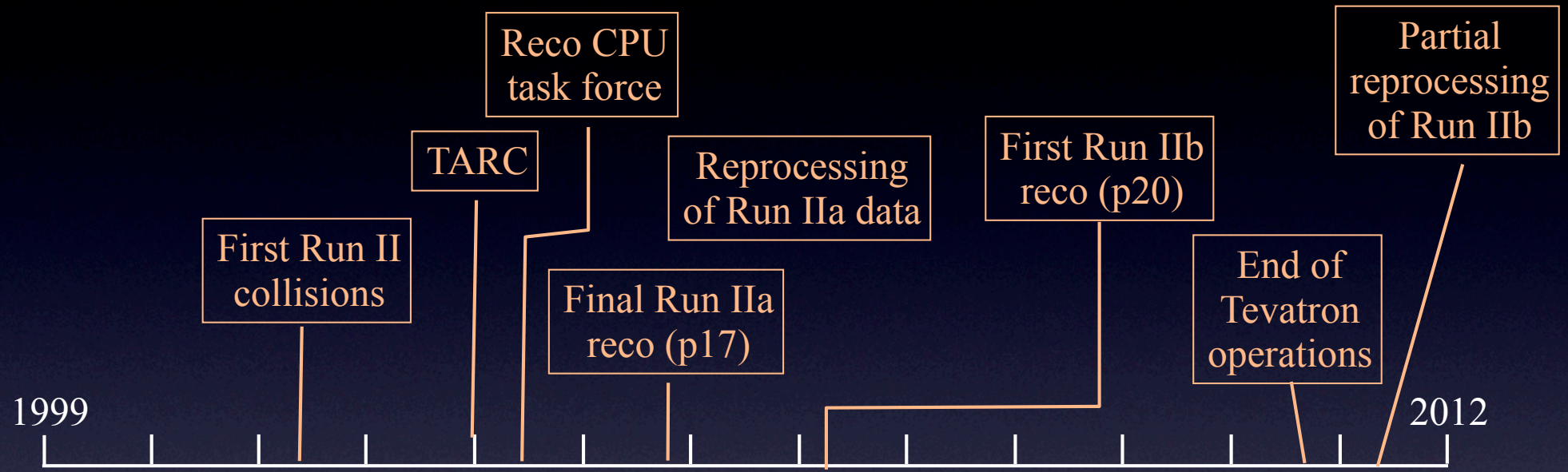
*except for S. Snyder

# Event Data Model

- Before we could even begin to write algorithms, needed to define a C++-compatible data format
  - no more ZEBRA!
- M. Paterno and S. Snyder led the effort
  - one issue was the need to convert objects created on the fly ("transient") to ones that could be stored ("persistent")
  - Scott developed an elegant solution, the DØ Object Module (DØOM)
- This work formed the basis upon which the Run II algorithms were built

# Run II Algorithms Timeline

## Milestones




S. Protopopescu
J. Womersley

H. Melanson
(S. Choi)

L. Duflot
(M. Hildreth)

A. Boehnlein
EWV

Algorithms/ computing combined

H. Greenlee
Q. Li

H. Melanson

H. Melanson
(L. Duflot)

M. Hildreth
(EWV)

EWV
Q. Li

A. Boehnlein
H. Melanson
H. Schellman

Elevated to directly below spokes

## Algorithm Coordinators

# Tracking

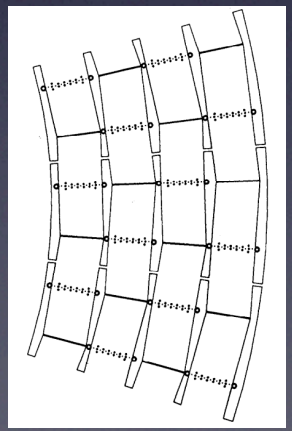
- From the Run I detector NIM paper

> The DØ detector was optimized with the following three general goals in mind:
> - Excellent identification and measurement of electrons and muons.
> - Good measurement of parton jets at large $p_T$ through highly segmented calorimetry with good energy resolution.
> - A well-controlled measure of missing transverse energy ($\not\!\!E_T$) as a means of signalling the presence of neutrinos and other non-interacting particles.

- As for central tracking:

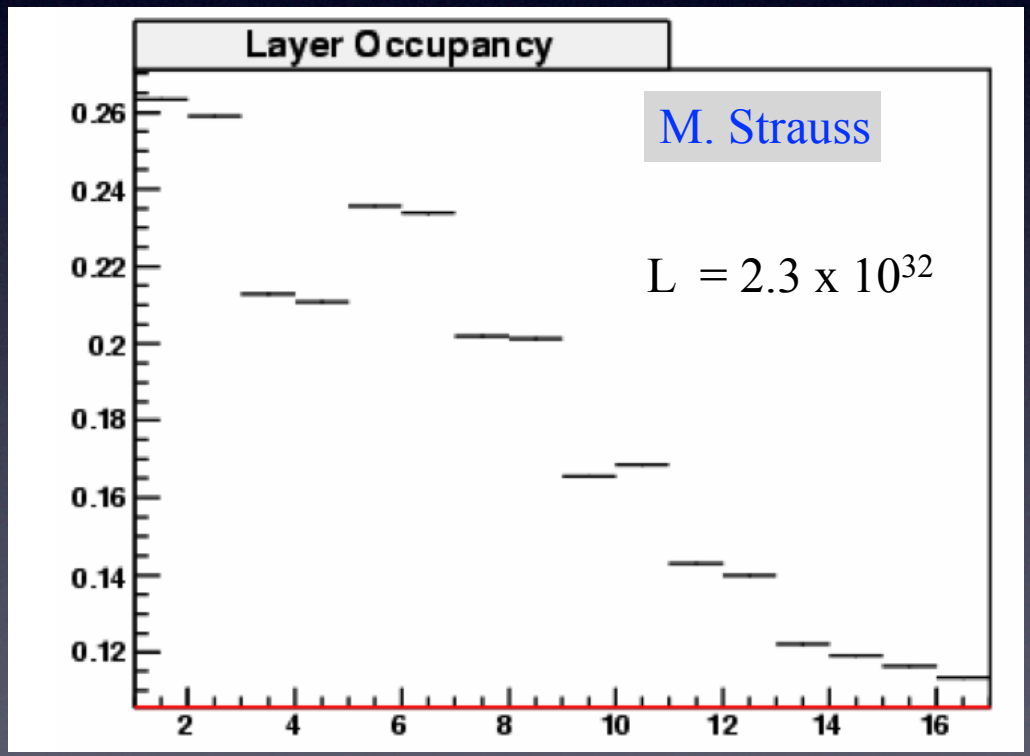
> - A compact non-magnetic tracking volume within $r = 75$ cm with adequate spatial resolution and particular emphasis on suppression of backgrounds to electrons.

- Still track reco in Run I was "easy"
  - no B field → straight tracks
  - low luminosity → low occupancy
  - 30 hits/track



# Tracking

- All of those advantages went away in Run II
  - solenoid added → even smaller tracking radius, curvature parameter in fits
  - high luminosity → high occupancy

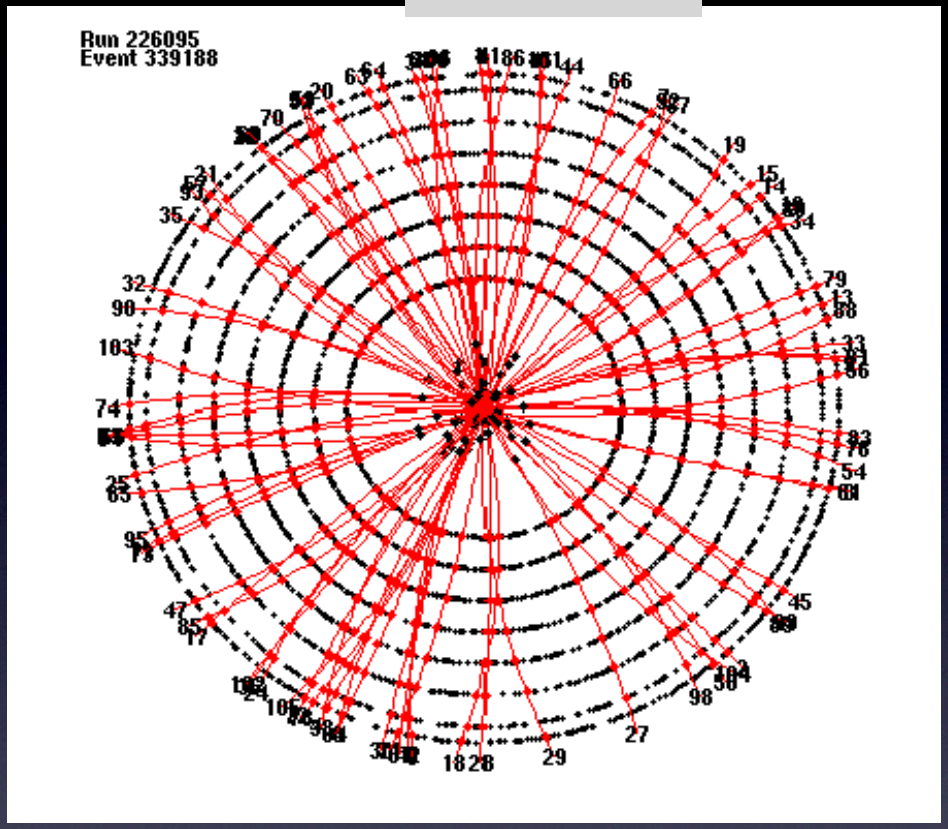



- Goals:
  - find all the real tracks
  - don't find too many "ghost tracks"
  - don't take too much CPU time

# Example of the challenge

- Level 3 trigger tracking tested on data event where all the CFT clusters were moved to random locations on their initial layers
  - 106 tracks found in the randomized event
- Only 85 tracks in the initial event!

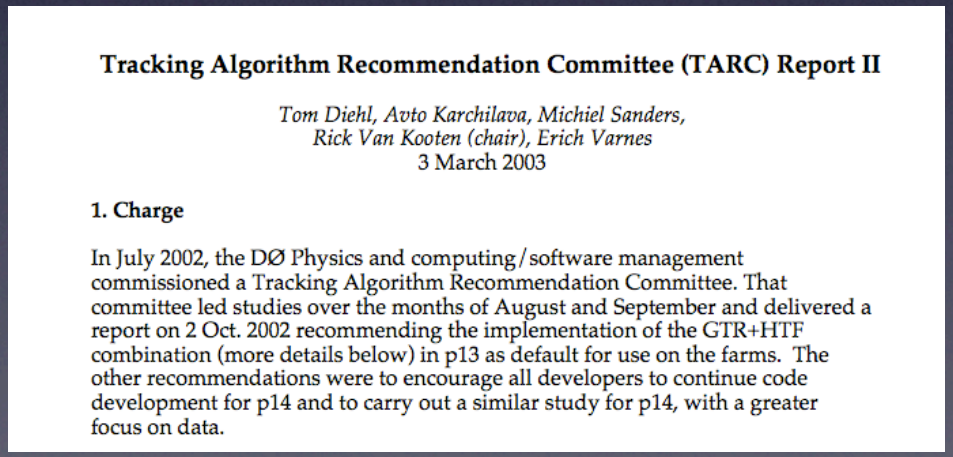
R. Bueselinck




"Fake" L3 event (hits from real event randomly distributed
106 tracks found -- 85 in real event

# TARC II

- When Run II data started coming in, became clear that tracking algorithm was not up to the challenge
- Tracking Algorithm Recommendation Committee charged with finding a solution
- Initial algorithm replaced with combination of
  - HTF (Hough Transform): S. Khanov
  - AA (Alternate Algorithm): G. Borissov
- Merging and final fitting of two set of tracks: H. Greenlee



**Tracking Algorithm Recommendation Committee (TARC) Report II**

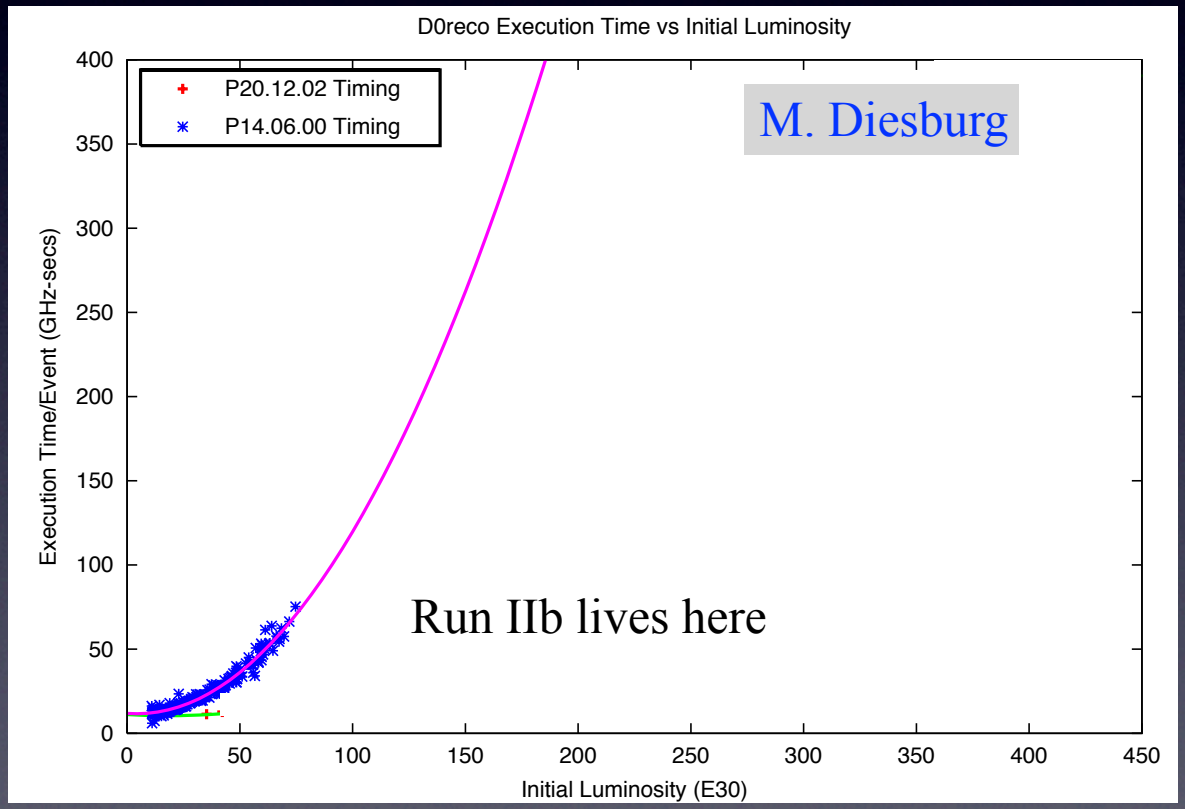
*Tom Diehl, Avto Karchilava, Michiel Sanders,*
*Rick Van Kooten (chair), Erich Varnes*
3 March 2003

**1. Charge**

In July 2002, the DØ Physics and computing/software management commissioned a Tracking Algorithm Recommendation Committee. That committee led studies over the months of August and September and delivered a report on 2 Oct. 2002 recommending the implementation of the GTR+HTF combination (more details below) in p13 as default for use on the farms. The other recommendations were to encourage all developers to continue code development for p14 and to carry out a similar study for p14, with a greater focus on data.

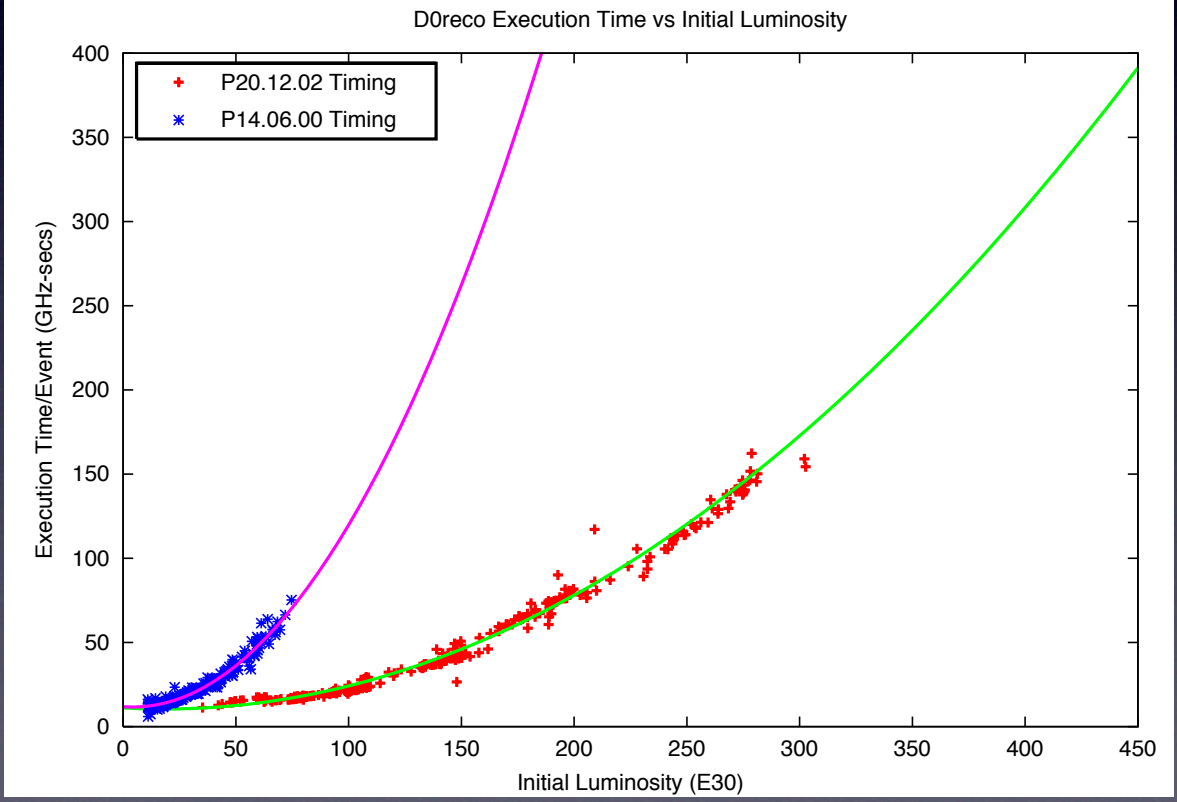
# DØReco CPU

- After TARC, the Run II reco algorithms were OK for Run IIa (low luminosity)
- But as the Tevatron improved, clear we were headed for trouble:



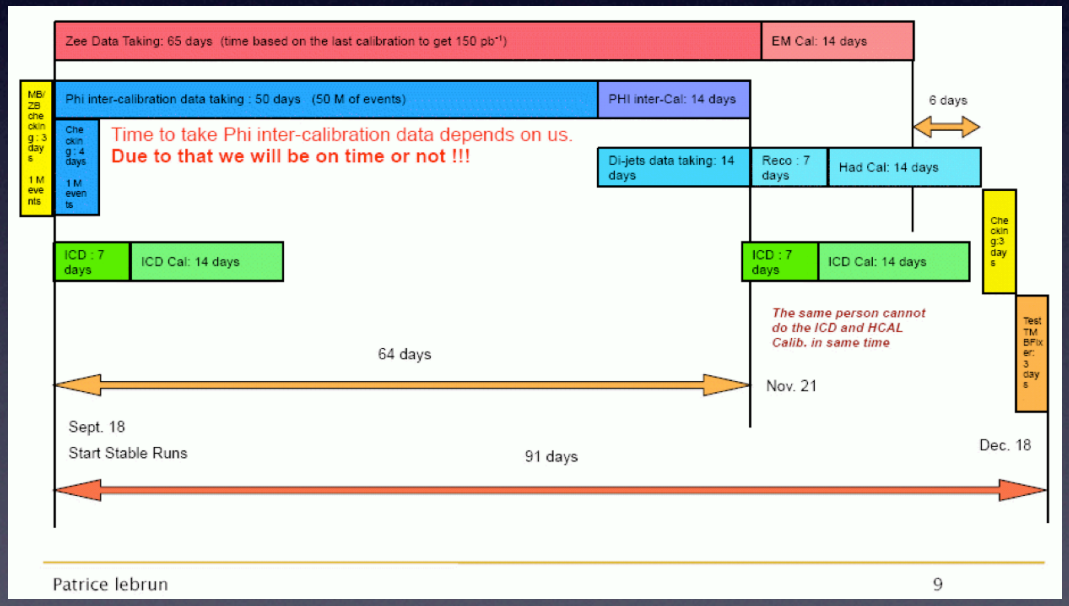
# Taming the CPU Monster

- Fitting d0reco into a realistic CPU budget involved
  - dedicated task force (chaired by Q. Li)
  - continual improvement to tracking and other algorithms
  - compromises (e.g. increase the minimum track $p_T$ threshold

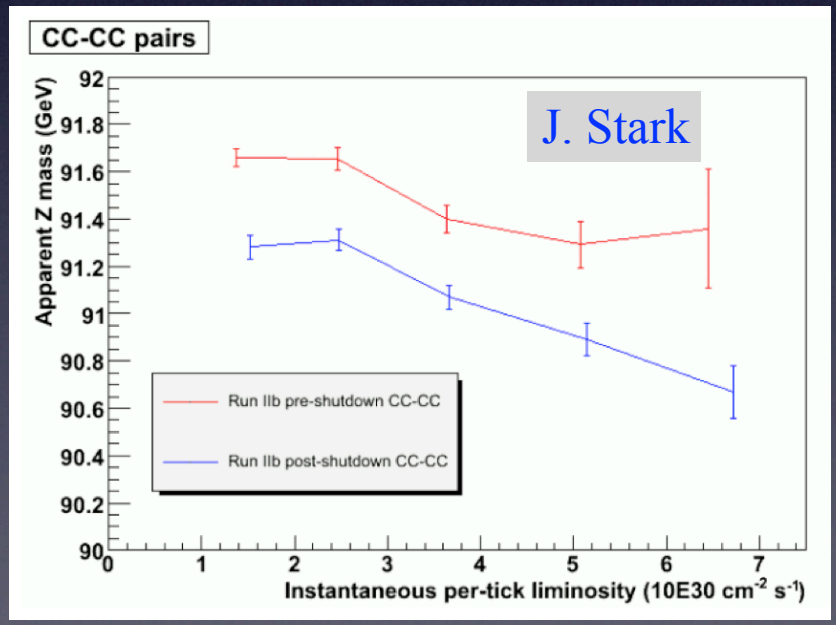


# Calorimeter Calibration

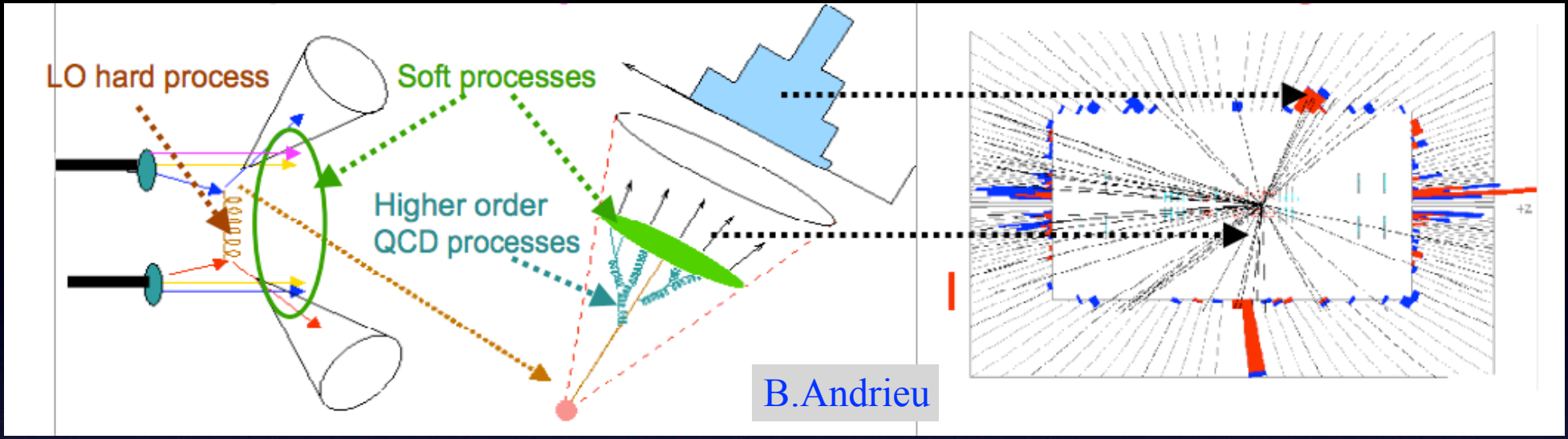
- DØ's U/LAr calorimeter is quite stable
- Nonetheless, periodically calibrations must be done to check response
- Complex process, involving special data triggers, analysis, reprocessing of data




Some interesting effects discovered....



# Jet Reconstruction




B.Andrieu

- Starting from Run I, DØ preferred cone algorithms
- Run II algorithm (mid-point cone) had better theoretical motivation than Run I
- Details (e.g. how often should jets be merged?) depend on conditions
  - more merging was allowed in Run IIa than the rest of the run (at higher luminosity)

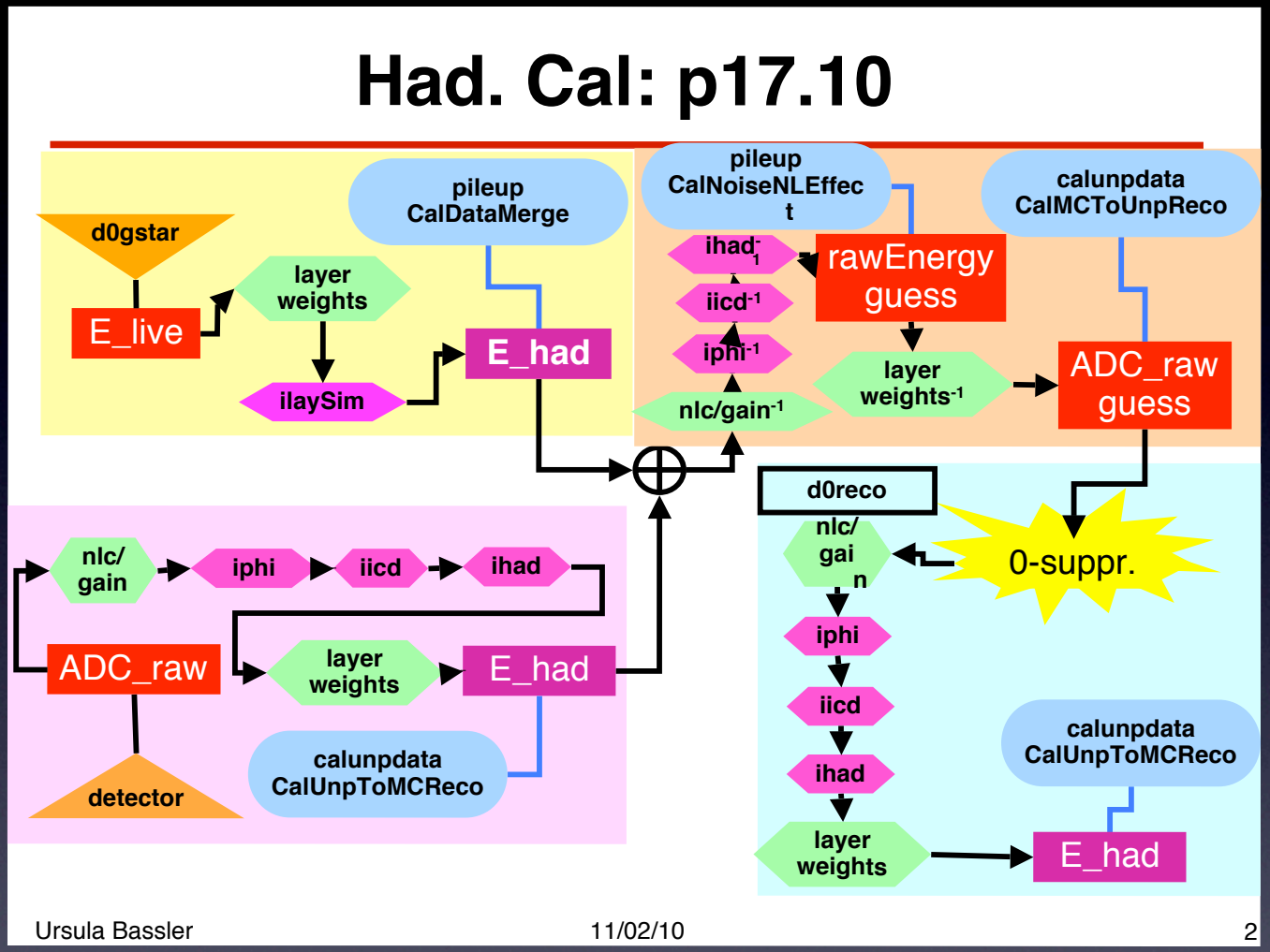
# Simulation

- Simulation is critical in HEP
- Starts with event generators that model the physics of our collisions
  - typically written by theorists
- Once the particles are produced and start interacting with DØ, it's up to us to provide the simulation
- Step one:
  - Simulate how the particles interact with the detector material (d0gstar)

S. Kunori

# Simulation

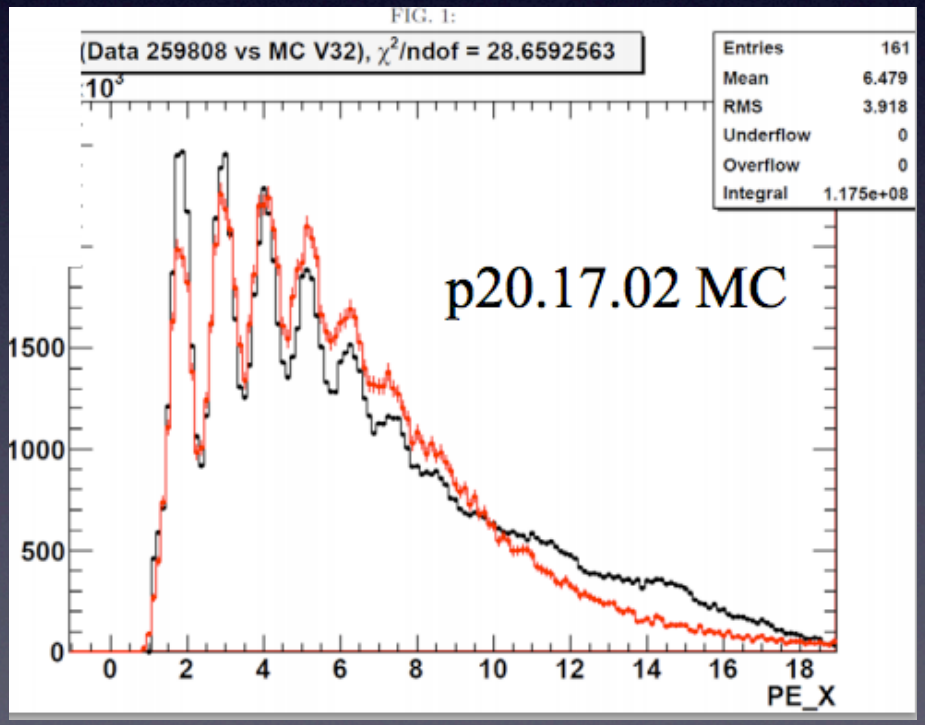
- Second step: simulated response of our electronics to signals (d0sim)
  - close cooperation between detector and software experts
- The better we do, the more the MC will look like the data
  - easier to tune analyses, smaller post-hoc scale factors
- One key improvement: use of real data events to model effects of multiple interactions/noise
  - more realistic than the previous procedure of adding some number of minbias MC events
- But it's complicated...




- This is just for the hadronic calorimeter...

# DØSim development

- DØsim continued to be refined throughout the run
- Some key developments:
  - use of per-sensor SMT efficiencies measured from data in MC (M. Aoki, H. Greenlee)
  - more accurate model of CFT response (G. Wilson)



# Object ID

- Initial versions for Run II:
  - S. Protopopescu (e/gamma), J. Womersley (jet/MET), D. Zieminska (muon), Q. Li (tau)
  - B ID was initially written to work on root trees rather than TMBs
    - one of the drivers that led to the development of CAF trees as the standard analysis format

# Object ID

- But object ID code is not the sort of thing you write once and forget
  - continual work to improve efficiency/better reject fakes
  - need to adapt algorithms for changing accelerator/detector conditions
  - need to measure performance in data and MC, and provide appropriate corrections
- Tremendous work by a dedicated (and usually small) group of people for each ID

# Summary

- DØ's physics output made possible by (among other things) highly capable core algorithms
- ~20 years of development
  - across drastic changes in computing, the detector, and accelerator conditions
- Constant effort to maintain performance at the highest possible level
  - the collaboration devoted a much smaller fraction of its manpower to this than the LHC experiments do
  - in many cases, heroic efforts by single individuals was crucial