Overview of Run 2 Analysis Software

- Software Infrastructure
- Event Display
- Reconstruction Software
- Monte Carlo Generation and Simulation
- Conclusions

Marjorie Shapiro
Feb 10, 1999

Software Infrastructure: Overview

- Goal is to provide a uniform environment for all non-Daq software:
  - Code Management and Distribution: CVS and SRT
  - Analysis Framework: AC++
  - Access to Event Data: Trybos/Root IO
  - Databases: Support Database API
  - Histograms, Ntupls, fitting: Root
  - Data Access: Modular, Exportable Data Handling System
- Allows easy migration of code from one application to another:
  - Online Consumer processes
  - Level 3 Trigger
  - Offline Reconstruction
  - Dataset Creation
  - On-site User Analysis
  - Off-site User Analysis

Software Infrastructure: Code Management and Distribution

- Code management via CVS (replaces Run 1 CMS repositories)
- Library and package building via SoftRelTools (SRT)
  - Set of Makefiles
  - List of package dependencies
  - OS specific setups

  CVS and SRT supported by FNAL CD. Has moved from development to maintenance phase.
- Daily updates of development release; frozen releases as required (roughly every 2 months at present).
- Code distribution occuring to many sites:

Nodes Receiving:	Linux	IRIX	OSF	Total
Frozen Release	38	8	4	50
Also, daily Development	20	4	3	27

  These numbers include the rebuilding platforms cdfsga, cdfpca and b0du02.

  Sun OS5 port in progress.
- Frozen releases distributed as prebuilt libraries and exe's as well as source code. Development incrementally updated (local build of libraries and exe's)
- Linux platform the most popular. Significant effort to make installation and distribution smooth and painless. Both network and CD-based installation supported.

See http://www-cdf.fnal.gov/offline/code_management/cdf_home_page.html)

Software Infrastructure: Analysis Control

- AC++ Design based on Run 1 AC
- Framework shared with Babar; separate local CVS repositories with code merges every ~ 6 months
- As in Run 1, code divided into "modules." AC++ provides flow control, IO control, etc
  - Run-time specification of which modules should be run and order of execution
  - Module parameters can be changed via Talk-to
  - Halting of event analysis in middle of path based on "filter" decisions by modules
  - Support for multiple output streams, with specification of which events/objects are written to each stream via path and filter decisions
- Some features still missing
  - Handling of histogram/ntuple files
  - Bank selection for output module
  - Build-Job++
  - Error signaling needs work

We have hired a computing professional to work with Liz Sexton on infrastructure software. He is starting to implement the necessary improvements to AC++

Software Infrastructure: I/O Interfaces

General agreement that Run 1 system had inadequate bookkeeping. We were not able to reproduce production results exactly. Fixing this problem is a major goal of the new system.

Requirement of reproducibility dictates that information flow through a limited number of well defined interfaces:

- **AbsParm:**
  Base class for Talk-to's. Allows setting of Module parameters such as cuts, thresholds, etc. Intend to provide journaling feature.
- **AbsEnv:**
  Global data of general interest (eg information that was stored in JOBSTA common block in Run 1)
- **Geometry Model:**
  Hierarchical tree that builds CDF geometry and associates it with material and dimensions. Tree initialized from CDF Geometry Database.
  Implementation still in progress
- **Support Database API:**
  Mechanism for fetching Calibration, alignment and other time-dependent physical parameters that can be retrieved from external databases (more to follow)
- **AbsEvent:**
  At present all event data stored in TryBos banks. Any module can add new banks to the event.
  Augmentation of Event Data Model in progress

Software Infrastructure: Support Databases

- Decide to use Relational Database rather than Run 1 Filebased system
- At Fermilab, need a high quality commercial database management system (DBMS)
  - Concurrent write access for multiple calibration users online
  - Robust backup and restore
  - Replication of tables between online and offline

  Oracle chosen as FNAL DBMS
- While Oracle may be appropriate for some off-site users, it is too heavy for many University groups
  - Hard to install and maintain
  - License fees

  Intend to provide mechanism to <u>export</u> tables for Oracle to freeware relational database (eg mSQL)
- Current database access routine ("support database API") has level of indirection necessary for hiding DBMS specific calls: Can swap between DMBS's
- Definition of table format done using simple definition language (IDL). Automatic code generation of SQL table definition, C++ header files, Oracle and mSQL specific access code

Software Infrastructure: Root

- Root is a Paw replacement writen by a team led by Rene Brun, the inventor of Paw
- C++ package that includes Paw functionality
  - Histograms
  - Ntuples
  - Fitting
  - Graphics
- But additional functionality as well
  - RootTuples can be much more general than ntuples- can write structures or objects instead of just a list of floats
  - Multistream IO: Can divide events into branches and put each branch in a separate file
  - Browsing of C++ classes
  - Debugger-like capabilities to call methods of classes
- Root is being widely used in HEP, but is still under development

Software Infrastructure: Root/continued

- Spring 1998, CDF decided to use Root I/O within Trybos (multiple IO streams)
- Sept 1998 recommendation of CDF/D0/CD Joint Working Group on Physics Analysis Tools:

  > We recommend that ROOT be adopted as the standard physics analysis package for Run II, contingent on a collaborative agreement with the ROOT team.

- CDF has begun serious design work on integration of Root
  - Fitting and Display for Online Calibration
  - Online Consumer Server
  - Geometry Database Prototype
  - Event Display Prototype
  - RootTuples and Histograms for analysis

  But, Isolate Root to insure that reconstruction and simulation code do not depend on Root classes

  Note: Paw will continue to be distributed as part of the Offline as long as CERN continues to maintain it. It is possible to write Hbook Ntuples from the Run 2 offline using a C++ interface package (Heptupl)

An Online Root Example: The Consumer Server

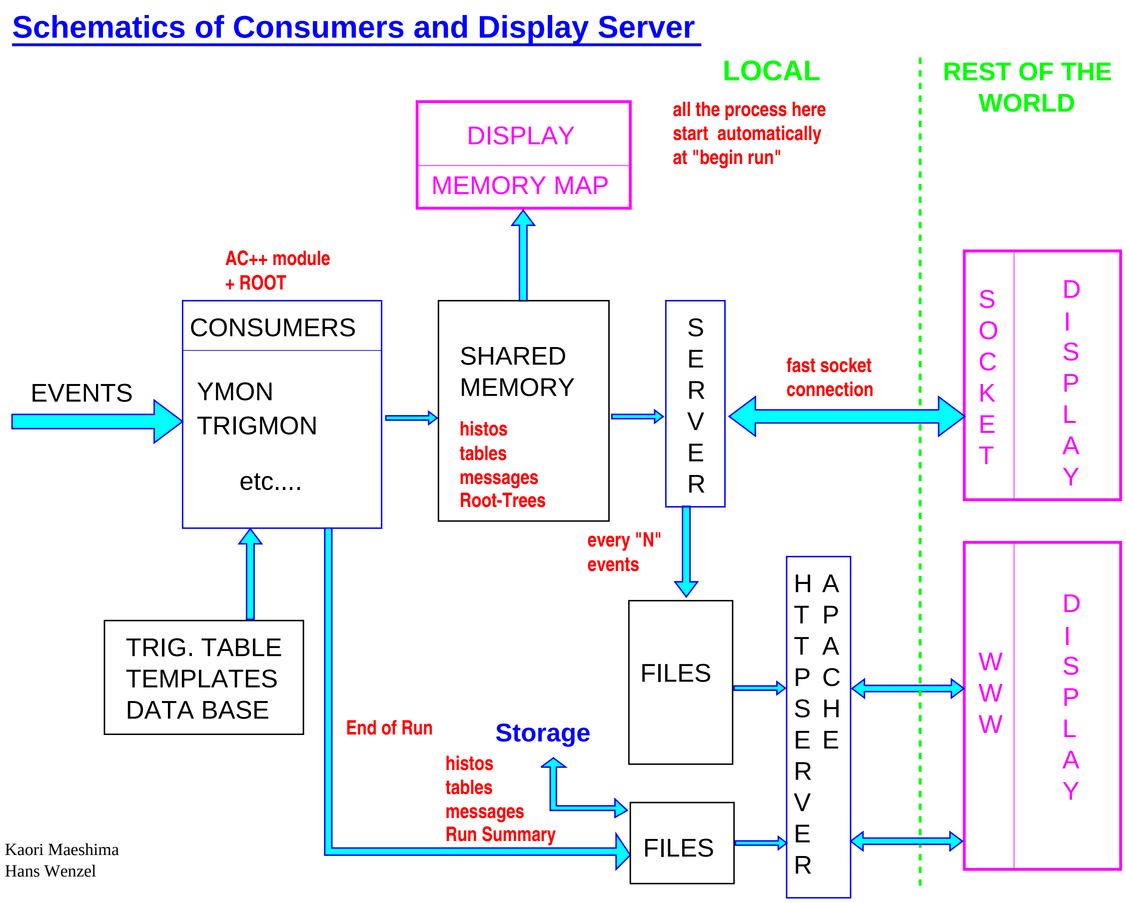


Figure 3: Schematic view of the proposed framework

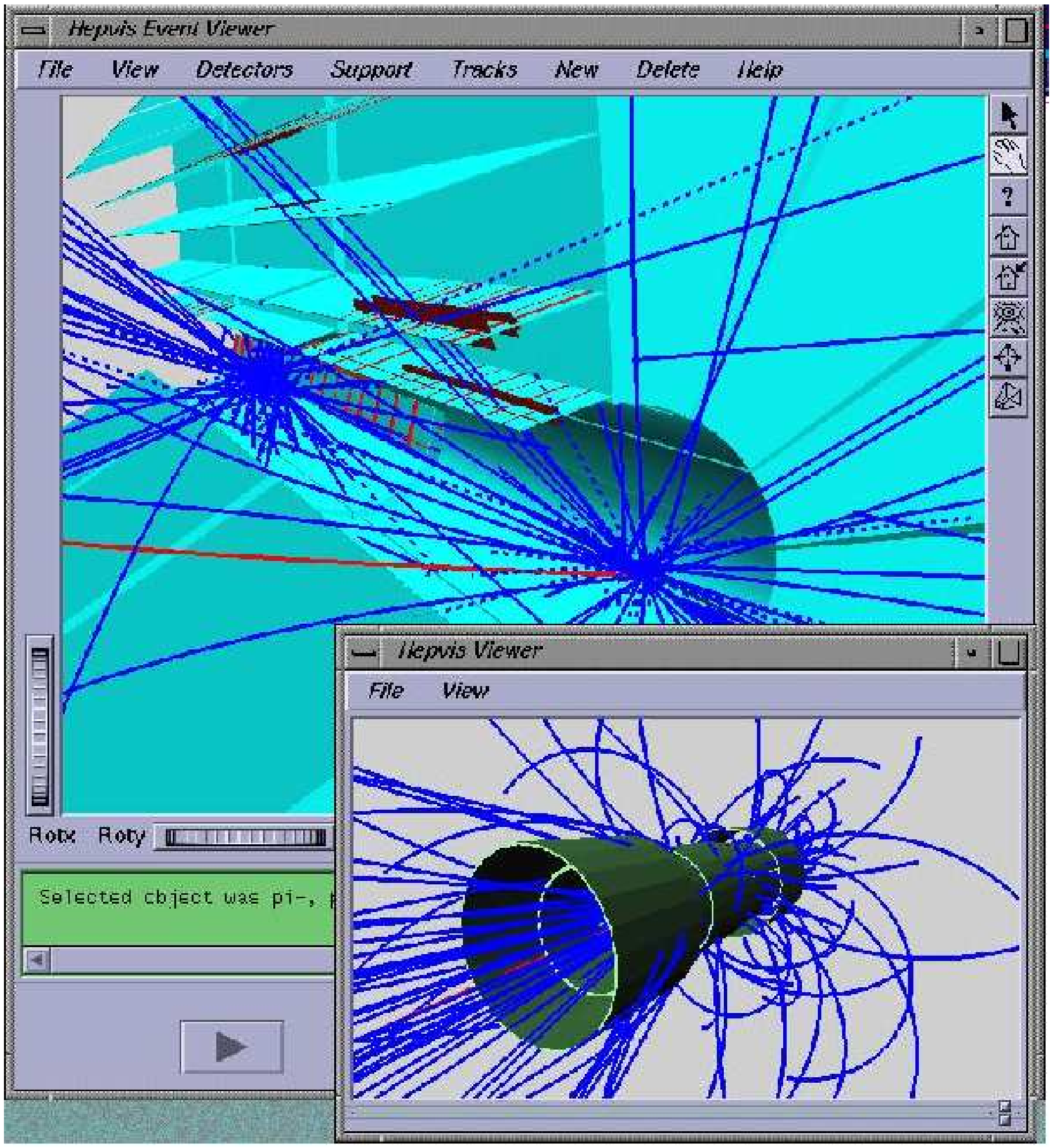
9

Kaori Maeshima
Hans Wenzel

Event Display

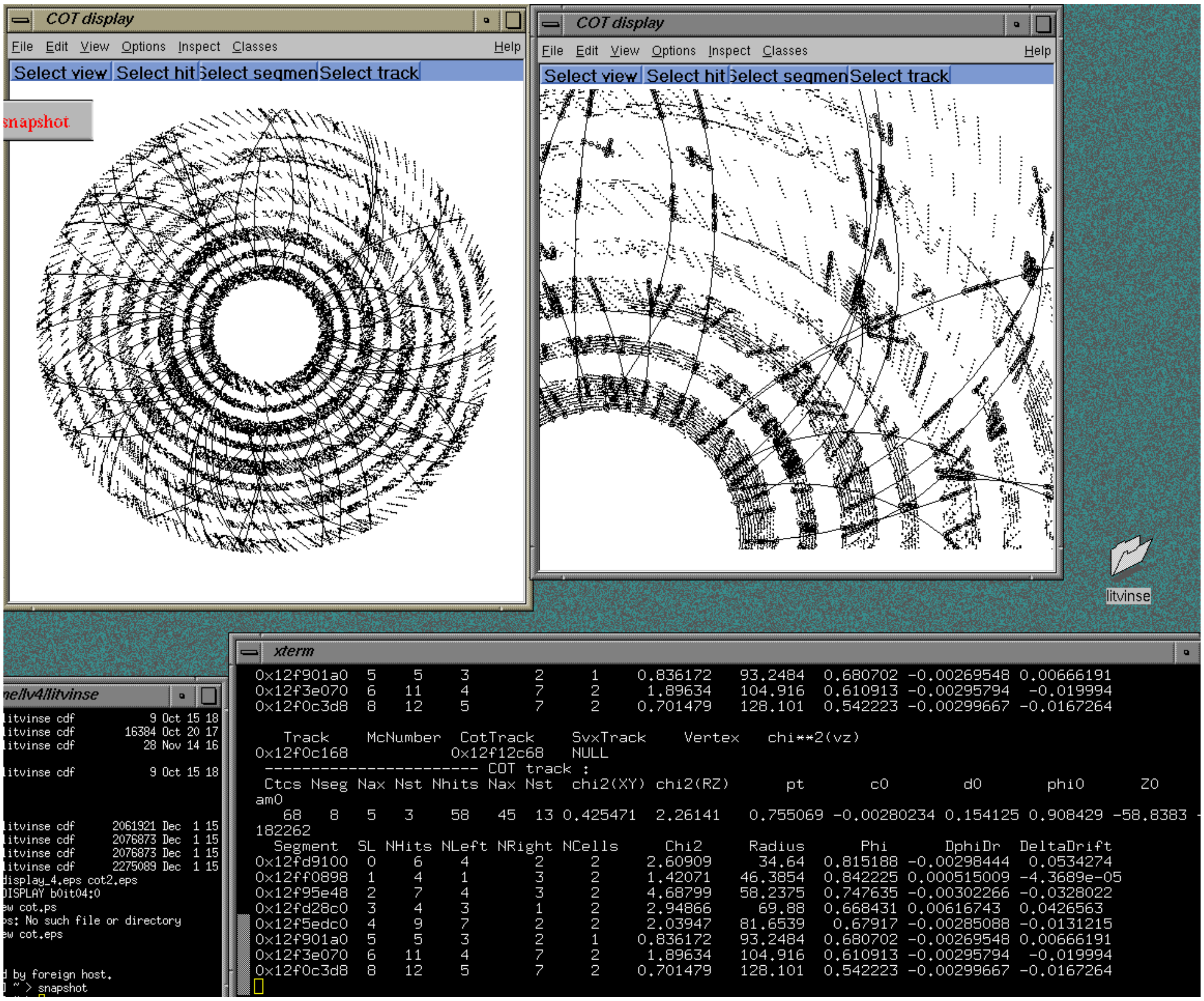
- Event display is the only infrastructure area where no final decision on underlying interface has yet been made
- Until Sept 1998, no CDF group was responsible for providing visualization framework.
  - ITEP-Moscow admitted to CDF with event display as an institutional responsibility
  - First task is to develop a proposal for unified display framework based on survey of collaboration's needs and on existing prototype work done both within the CDF collaboration and by the ITEP group
- Prototypes now exist within two frameworks:
  - OpenInventor
    - High quality 3D graphics
    - HepVis package provides easy interface
    - CD support and expertise
    - Difficult to merge with AC++
    - Requires license negotation (via JP)
  - Root
    - Supports 2D and 3D graphics classes
    - Natural match to data model and physics analysis tools
    - Runs within AC++
    - Monolithic: Must understand implications

Open Inventor Event Display



- Complete package for silicon tracker, including hits, tracks, Monte Carlo Truth
- Valuable tool for debugging tracking algorithms
- Heavy use of rotation and zoom features by code developers

Root Event Display



- Silicon and Drift Chamber displays exist
- Runs within AC++ framework
- Pulldown menus for view and hit selection
- Allows access to RootCint prompt

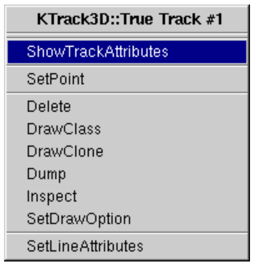
Root Event Display/cont

Another Root example

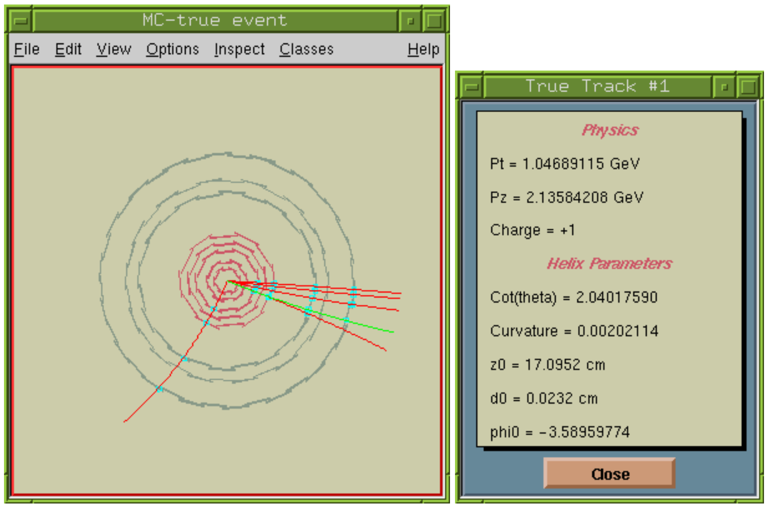
UNIVERSITY OF KARLSRUHE

Track Attributes

A new class is used for the graphical representation of tracks in the event display. Attributes of drawn tracks are now accessible interactively via the context menu of this new class KTrack3D:



The track's name is True Track #1, indicating it is not a reconstructed, but a Monte Carlo true track. The new class KTrack3D inherits from the ROOT standard class TPolyLine3D. For the time being the only additional method implemented is ShowTrackAttributes. When the corresponding item in the context menu is chosen, a window showing the track's properties pops up, as one can see in the example below.



Reconstruction Software

- Basic architecture of reconstruction similar to Run 1
- Code divided into modules that:
  - Map raw data into hits: "DtoE"
  - Perform pattern recognition on the hits within a single detector: "EtoS," clustering
  - Combine results for different detectors (match track segments between silicon and drift chamber, match tracks to calorimeter clusters or muon stubs, etc)
  - Identify reconstructed objects as potential physics objects (electrons, muons, jets, τ's,...)
- Code developers allowed to choose between Fortram and C++. All developers have chosen C++ to write new code. However, we continue to use legacy Fortran when appropriate:
  - Substitute for C++ until new code is available
  - Verify new code by comparing results to those obtained from Run 1 Fortran
  - Provide guide as to potential performance (speed and efficiency)
  - Retain knowledge of "real-life" details and examples of how to cope with these details

Reconstruction Status and Plans

- Oct 1998 Milestone: 1st pass of a complete Production executable. Goals of this pass:
  - To set overall "vision" of reconstruction package
  - To establish interaction among analysis modules
  - To provide prototype to be used for evaluation of overall event data model
- Expected many modules to have simplified algorithms, but first opportunity to study performance
- Special emphasis placed on some packages:
  - Tracking: Most labor and most CPU intensive
  - Jets: overall design structure for calorimeter code
- Most modules met the Oct milestone. Exceptions:
  - Central muon stub finding
  - CPR DtoE
  - Plug preradiator and strip chamber DtoE and cluster finding
  - Plug electron finding
- Milestone anticipated use of legacy Fortran for CES clustering, electron finder and muon-track matching code. These modules will be replaced with new code this Spring.

Comments on Reconstruction:

- Basic control and data flow are fine: no serious redesign needed
- Excellent progress on tracking, calorimetry and jets. Validation with Run 1 data looks good.
- Redesign of electron and muon stub finders has started
- Muon-track linking critically dependent on both track extrapolation class and geometry description. Serious design work will begin in March, after new geometry classes have been finalized.
- Next integration milestone: May 1998
  - Final calibration API and real database
  - Final geometry model
  - New versions of muon stub finding, electron finding, muon-track linking

Name	Description	Status
Calorimetry	Calibrates raw data, formats data into collection of towers	1^{st} version exists with dummy calibration consts
SiClustering	Calibrates raw data, creates collection of clusters	1^{st} version exists with dummy calibration consts
PVFinder	Finds primary vertex from Silicon	1^{st} version exists
SiPatternRec	Silicon pattern recognition and fit	1^{st} version exists
CT_TrackingModule	COT pattern recognition and fit	1^{st} version exists
OITrack	Associate Silicon hits with existing COT tracks	1^{st} version exists
SvxCotTrackFit	Merge Silicon and COT tracks	1^{st} version exists
Jetclu	Jet Finder	exists
MetModule	Calculates Missing E_T	1^{st} version exists
emclst	Find electromagnetic clusters	1^{st} version exists (central calor only)
Stpana	Strip Chamber calibration and clustereing	Fortran with dummy calibration
ElctrnModule	Match electromagnetic clusters to strip chambers and tracks	Fortran (central only) C++ rewrite in progress
CMXMods_DtoE	CMX calibration and creation of muon hit	1^{st} version exists dummy calibration only
CMUMods_DtoE	CMU calibration and creation of muon hits	1^{st} version exists with dummy calibration only
CMPMods_DtoE	CMP calibration and creations of muon hits	1^{st} version exists with dummy calibration only
CSXMods_DtoE	Muon scintillator calibration and creation of muon hits	1^{st} version exists with dummy calibration only
IMUDtoE	Intermediate Muon calibration and creation of muon hits	1^{st} version exists with dummy calibration
CMEtoS	Create cental muon stubs from hits	work in progress
IMUstubs	Create IMU stubs from hits	1^{st} version exists
CMLnkModule	Link muon stubs to Tracks	Fortran C++ redesign in progress
TauModule	Search for τ candidates	Not started
dEdxModule	Calculate particle id	Not started
Luminosity	Calculate luminosity of production dataset	Not started
JetcluModule	Run 1 Jetfinder	Fortran (for comparisons only)
EMclstModule	Run 1 Em clustering	Fortran (for comparisons only)

Table 1: Status of Offline Reconstruction Modules

Physics Generators

- Use CD distributions of standard generators and STDHEP interface to HEPEVT common block
- Single CDF specific routine to move STDHEP to Trybos
- Currently supported with Run 2/AC++ Framework:
  - Herwig V5.9
  - Isajet V7.31
  - Pythia V6.1
  - MinBias (update of MBR)
  - FakeEv (Single Particle Gun)
  - QQ (Cleo Decay package)
  - Event Mixing (Poisson distribution of min bias added to generated event)
  - Vecbos
- Future Work:
  - Random Number management
  - Addition of more exact matrix element generators
  - Herprt

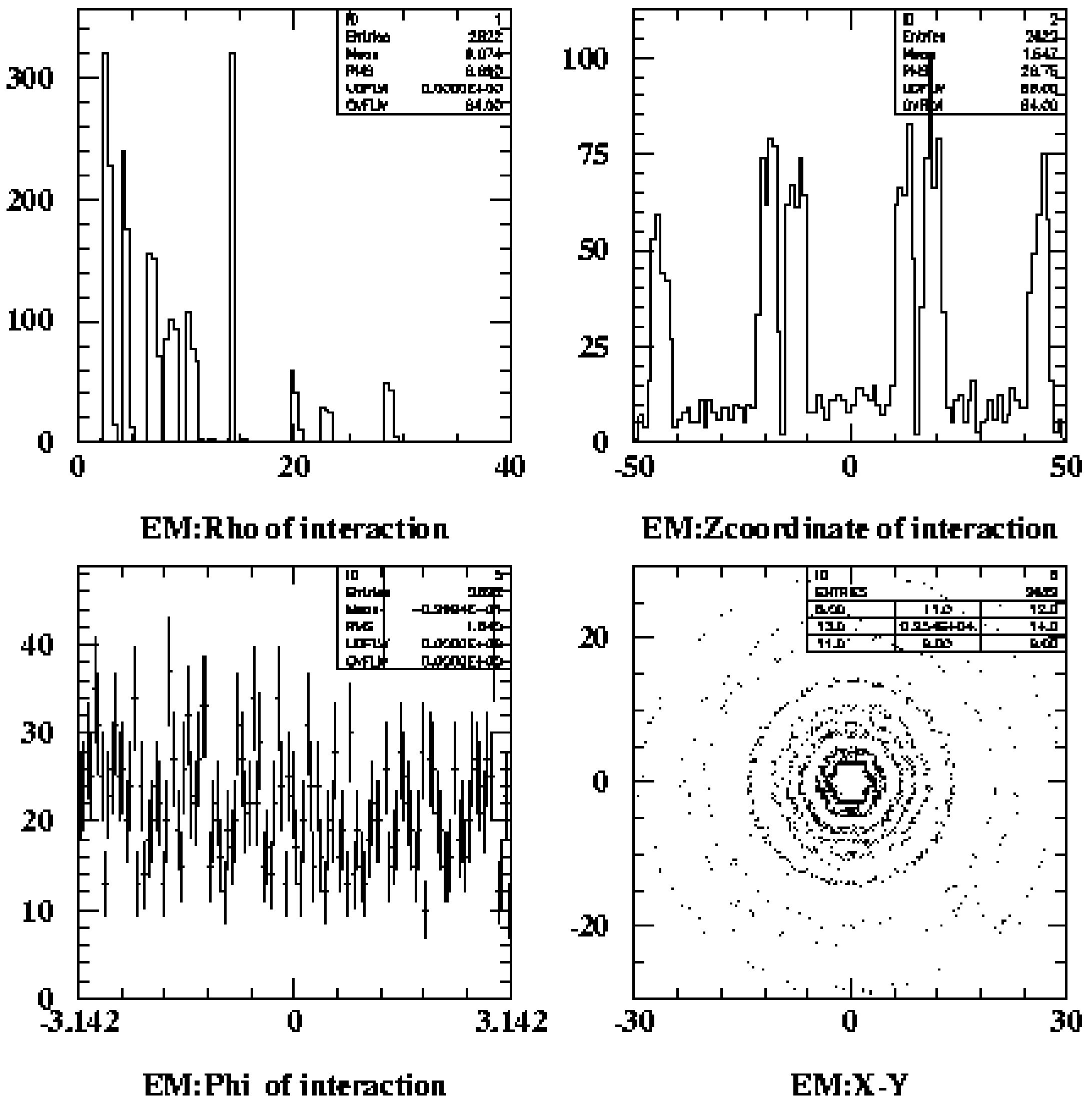
Geant Simulation

- Simulation available since Feb 1998. Based on Geant 3.21
- Geant 3 implementation hidden from CDF developers by C++ adaptor classes:
  - Allows movement to Geant 4 in future
  - Allows development of common geometry and material description for use with reconstruction as well as simulation
  - Provides management and control layer to allow run-time switching between different simulation modes with different levels of detail in both geometric description and digitization.
- No use of Geant 3 hits and digits; all hit and digitization code CDF specific

Recent Developments in Simulation:

- Improved Talik-to interface to Geant 3 (cutoffs, etc)
- Improved modeling of dead material
- New Trybos banks to store links between hits and Monte Carlo tracks
- Original parameterized shower option (QFL) in calorimetry augmented by
  - Very detailed description of plug calorimeter for use in tuning parametrized response
  - Gflash description of hit generation (better description of transverse spreading than the home-grown parameterizations)
- Garfield studes of drift time in central tracker used to parameterize hit time and shape

Material Distributions in Silicon Tracker



X-ray material using conversions

Example of Detailed Geometry Description of Plug Calorimeter

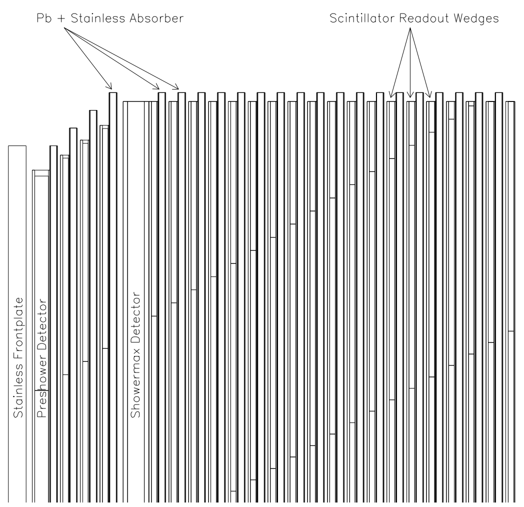


Figure 2: The outer edge of the EM calorimeter as modeled with GEANT. A stainless steel front plate is followed by 22 Pb+Stainless absorber plates. Also shown is the location of the scintillator readout wedges, the SMD and the PSD.

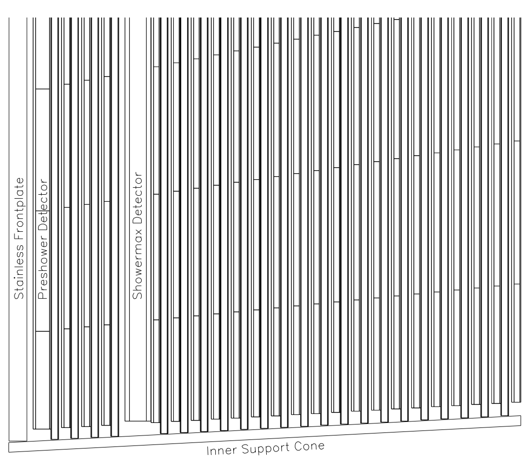


Figure 3: The details of the EM calorimeter inner edge as modeled with GEANT. The edge follows a 3° support cone extending from the front plate to the rear of the compartment.

Comparison of Hadronic Resolution of Geant Simulation and Test Beam

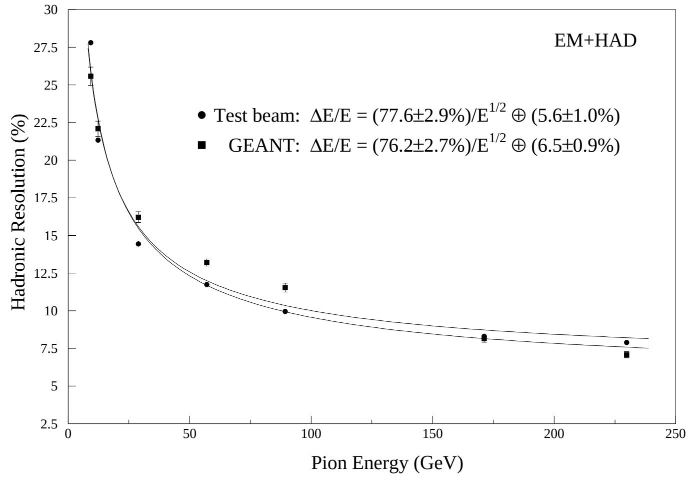


Figure 11: The simulated hadronic energy resolution of the Plugcal compared with test beam results ([5]) using the "EM+HAD" calibration scheme outlined in Section 3.2.1.

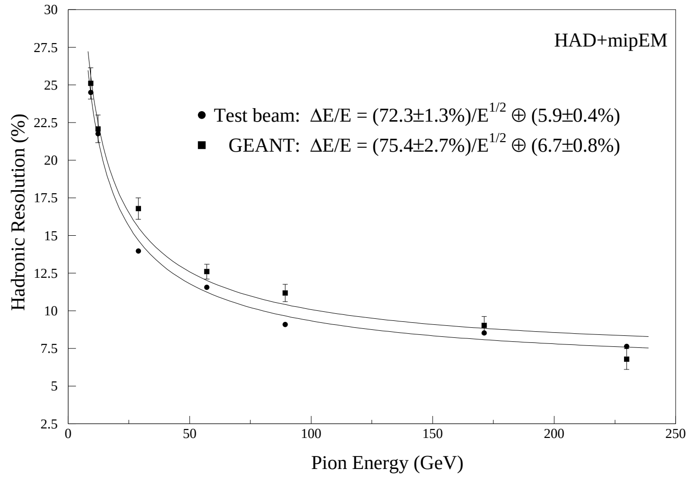


Figure 12: The simulated hadronic energy resolution of the Plugcal compared with test beam results ([5]) using the "HAD+mipEM" calibration scheme outlined in Section 3.2.2.

10

Conclusions

- Basic infrastructure software exists
- Code management and distribution allow remote access and remote development
  - Linux OS in best shape
- Support databases in Oracle on-site with export to mSQL for off-site use planned
- Baseline version of event generators and simulation exist
- First pass production exe exists, but still under development
- All areas still need documentation, packaging, cleanup for use by the "average CDF physicist"

Expect a "user friendly" release (with documentation) by June