



# DØ Calibration Database Browser

## *Design and Implementation*

Henry Barnor

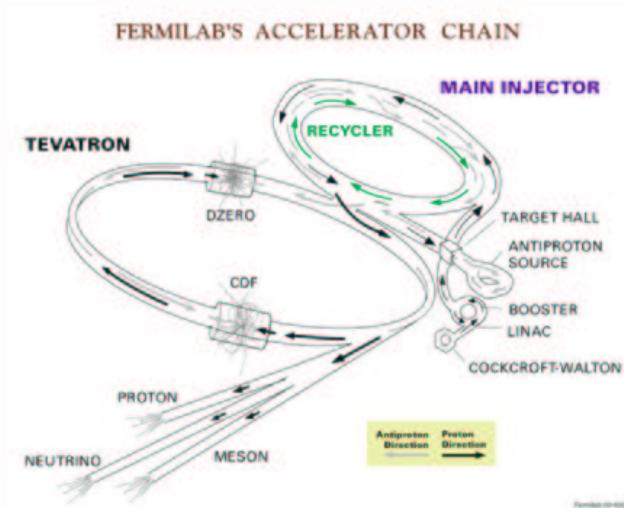
hbarnor@fnal.gov

DØ



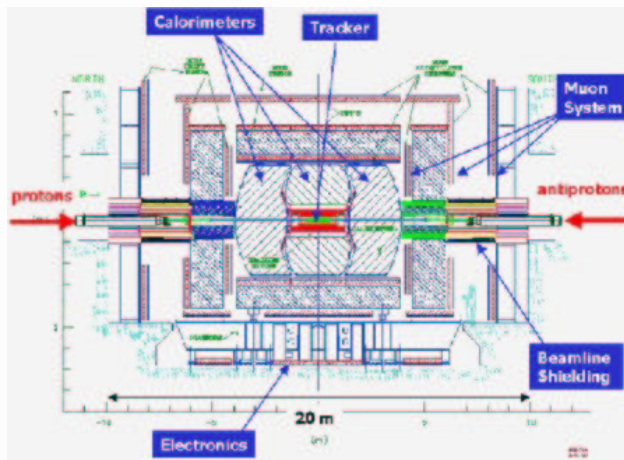
# Overview

✓ What is DØ ?



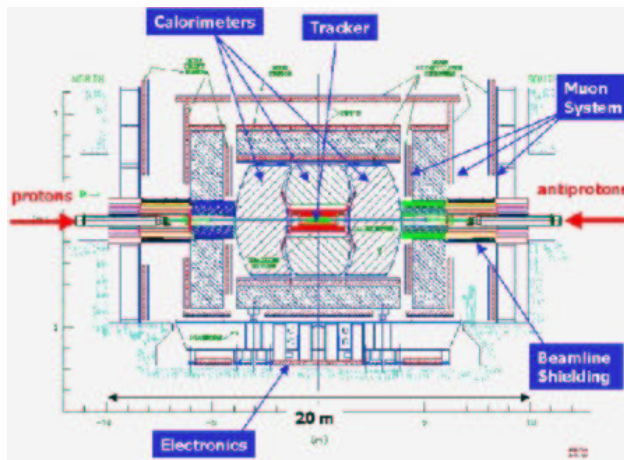
# Overview

- ✓ What is  $D\emptyset$  ?
- ✓ Two tracking systems



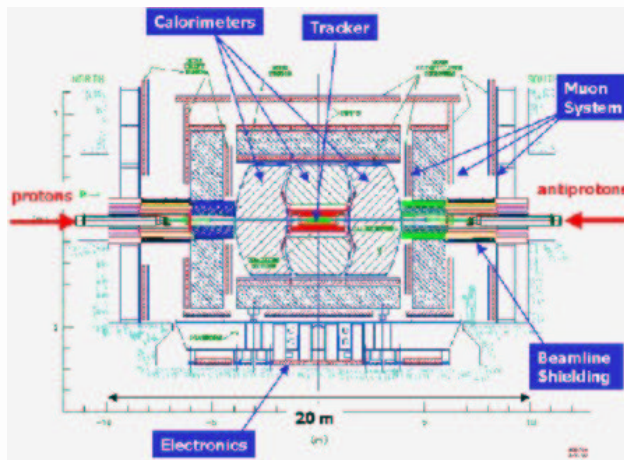
# Overview

- ✓ What is  $D\emptyset$  ?
- ✓ Two tracking systems
- ✓ The Silicon Microstrip Tracker and the Central Fiber Tracker



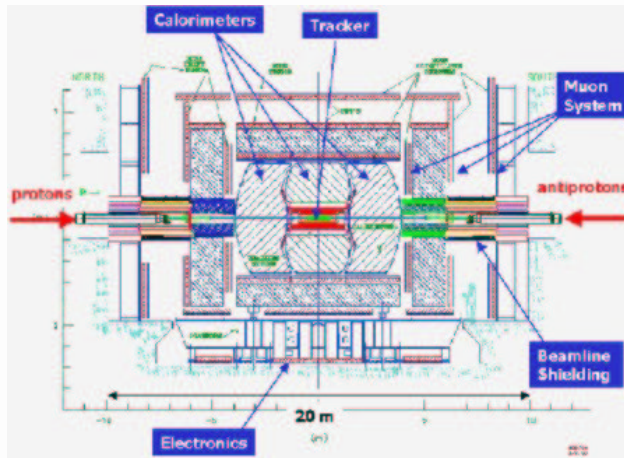
# Overview

- ✓ What is  $D\emptyset$  ?
- ✓ Two tracking systems
- ✓ The Silicon Microstrip Tracker and the Central Fiber Tracker
- ✓ Calorimeters are used to capture the electrons and hadrons(quarks)

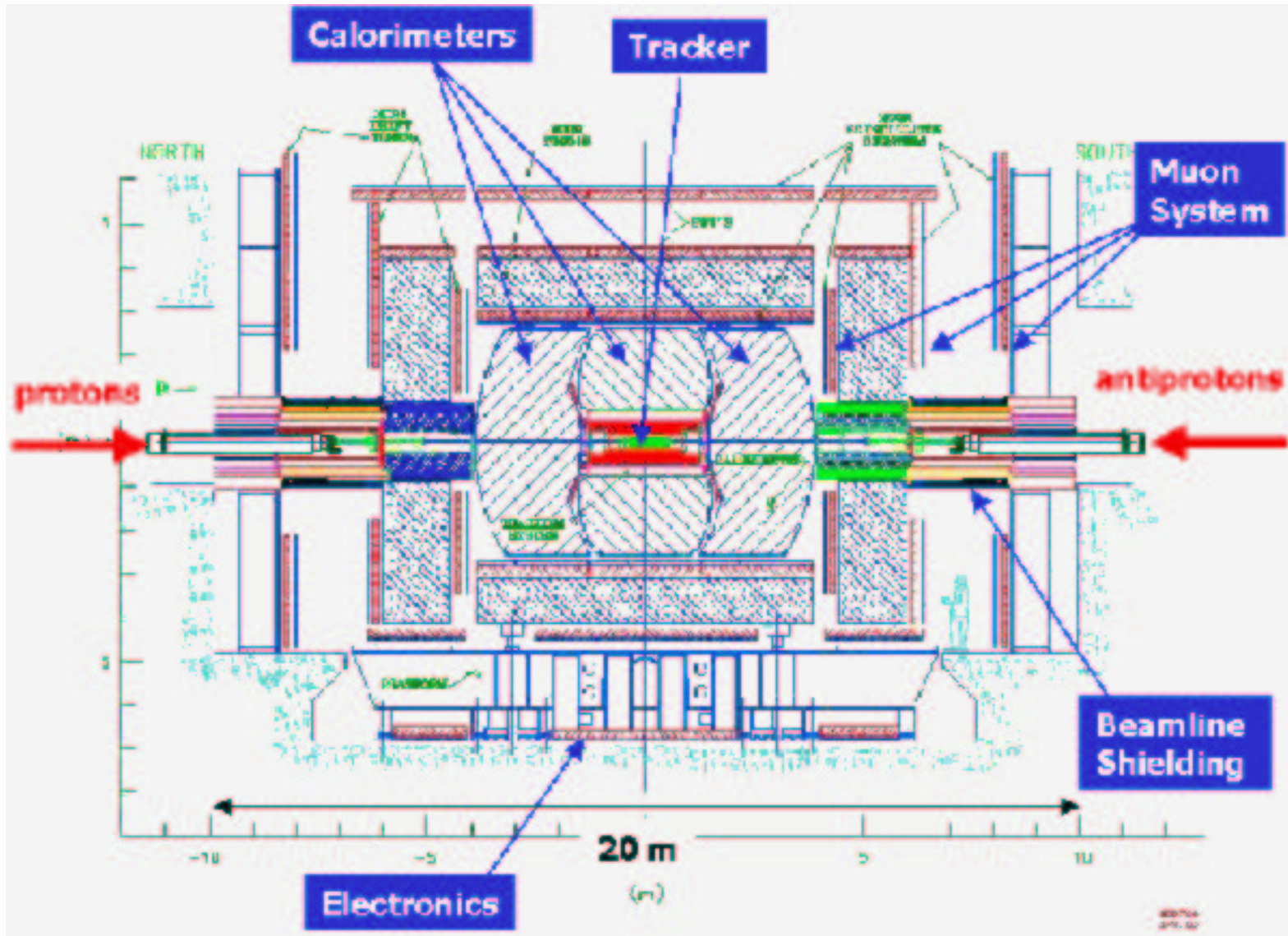


# Overview

- ✓ What is  $D\emptyset$  ?
- ✓ Two tracking systems
- ✓ The Silicon Microstrip Tracker and the Central Fiber Tracker
- ✓ Calorimeters are used to capture the electrons and hadrons(quarks)
- ✓ Wire chambers outside the calorimeter capture muon



# The DØ Detector



# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.



# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.
- ✓ The signal is amplified and digitized



# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.
- ✓ The signal is amplified and digitized
- ✓ The energy is then calculated using

$$E = aV + b$$



# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.
- ✓ The signal is amplified and digitized
- ✓ The energy is then calculated using

$$E = aV + b$$

- ✓  $a$  = multiplication factor(gain)  
 $b$  = offset(pedestal)



# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.
- ✓ The signal is amplified and digitized
- ✓ The energy is then calculated using

$$E = aV + b$$

- ✓  $a$  = multiplication factor(gain)  
 $b$  = offset(pedestal)
- ✓ Obtaining the above constants is called calibration



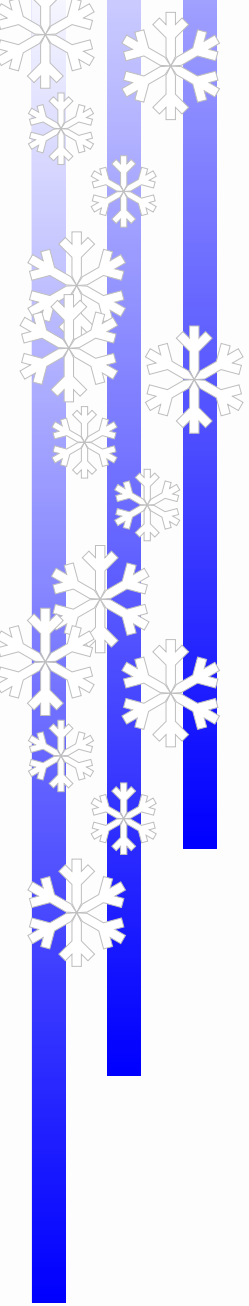
# Calibration

- ✓ Detectors work by measuring ionization as electrical signal.
- ✓ The signal is amplified and digitized
- ✓ The energy is then calculated using

$$E = aV + b$$

- ✓  $a$  = multiplication factor(gain)  
 $b$  = offset(pedestal)
- ✓ Obtaining the above constants is called calibration
- ✓ Oracle database is used to store the gain and pedestal values





# My contribution to Fermilab



# Database Browser Design

- ✓ My project - A graphical browser for the calibration database



# Database Browser Design

- ✓ My project - A graphical browser for the calibration database
- ✓ Should be web based



# Database Browser Design

- ✓ My project - A graphical browser for the calibration database
- ✓ Should be web based
- ✓ Allow for searching



# Database Browser Design

- ✓ My project - A graphical browser for the calibration database
- ✓ Should be web based
- ✓ Allow for searching
- ✓ Should be able to plot data



# Database Browser Design

- ✓ My project - A graphical browser for the calibration database
- ✓ Should be web based
- ✓ Allow for searching
- ✓ Should be able to plot data
- ✓ Accept SQL queries from advanced users



# First Implementation

✓ MISWEB



# First Implementation

- ✓ MISWEB
- ✓ A web-based tool for displaying data from an Oracle database.



# First Implementation

- ✓ MISWEB
- ✓ A web-based tool for displaying data from an Oracle database.
- ✓ Problems with MISWEB



# First Implementation

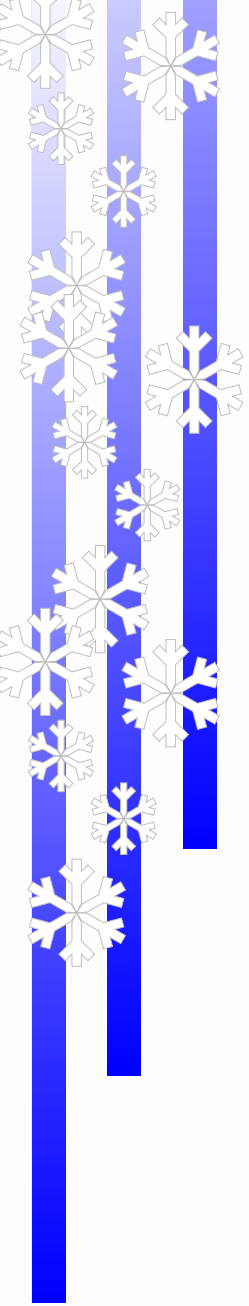
- ✓ MISWEB
- ✓ A web-based tool for displaying data from an Oracle database.
- ✓ Problems with MISWEB
- ✓ Cannot explore in more than one direction



# First Implementation

- ✓ MISWEB
- ✓ A web-based tool for displaying data from an Oracle database.
- ✓ Problems with MISWEB
- ✓ Cannot explore in more than one direction
- ✓ No plotting capabilities





**How do we counter these limitations ?**



# Second Implementation

- ✓ Start from scratch



# Second Implementation

- ✓ Start from scratch
- ✓ OOP - flexible, robust, extensible, easy to maintain



# Second Implementation

- ✓ Start from scratch
- ✓ OOP - flexible, robust, extensible, easy to maintain
- ✓ JFreeChart - open source, free plotting library



# Second Implementation

- ✓ Start from scratch
- ✓ OOP - flexible, robust, extensible, easy to maintain
- ✓ JFreeChart - open source, free plotting library
- ✓ Java Applet



# Second Implementation

- ✓ Start from scratch
- ✓ OOP - flexible, robust, extensible, easy to maintain
- ✓ JFreeChart - open source, free plotting library
- ✓ Java Applet
- ✓ Model-View-Controller(MVC) Design pattern



# cont. Second Implementation

✓ Final Product



# cont. Second Implementation

- ✓ Final Product
- ✓ Problems



# cont. Second Implementation

- ✓ Final Product
- ✓ Problems
- ✓ Security sandbox



# cont. Second Implementation

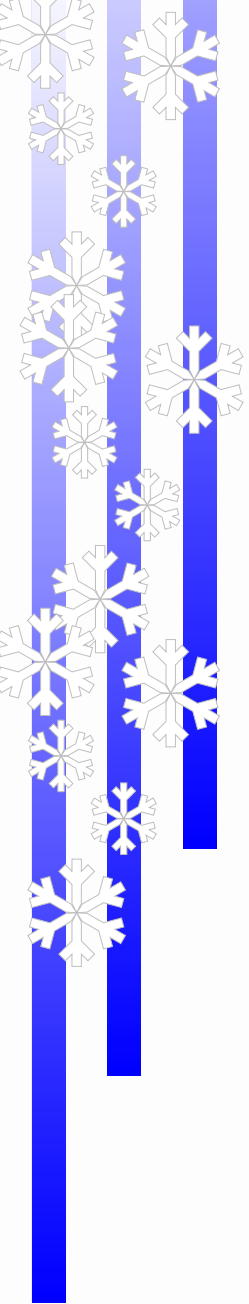
- ✓ Final Product
- ✓ Problems
- ✓ Security sandbox
- ✓ Signed jar



# cont. Second Implementation

- ✓ Final Product
- ✓ Problems
- ✓ Security sandbox
- ✓ Signed jar
- ✓ Solution: Reimplement using Server-side technology





**Do it all over again ?**



# Third Implementation

✓ OOP saves the day



# Third Implementation

- ✓ OOP saves the day
- ✓ MVC - loose coupling allows replacement of the view



# Third Implementation

- ✓ OOP saves the day
- ✓ MVC - loose coupling allows replacement of the view
- ✓ HTML, Javascript, Java Server Pages(JSP) and Java Servlets



# Third Implementation

- ✓ OOP saves the day
- ✓ MVC - loose coupling allows replacement of the view
- ✓ HTML, Javascript, Java Server Pages(JSP) and Java Servlets
- ✓ Extend the controller to deal with servlets



# Third Implementation

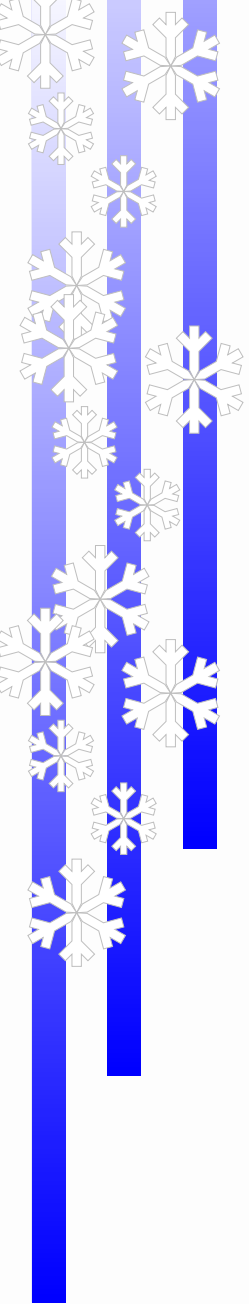
- ✓ OOP saves the day
- ✓ MVC - loose coupling allows replacement of the view
- ✓ HTML, Javascript, Java Server Pages(JSP) and Java Servlets
- ✓ Extend the controller to deal with servlets
- ✓ Use Java Server Pages to create a view



# Third Implementation

- ✓ OOP saves the day
- ✓ MVC - loose coupling allows replacement of the view
- ✓ HTML, Javascript, Java Server Pages(JSP) and Java Servlets
- ✓ Extend the controller to deal with servlets
- ✓ Use Java Server Pages to create a view
- ✓ Another final product





**Why i should be rehired next  
summer.**



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success
- ✓ Java server-side - Successful but ...



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success
- ✓ Java server-side - Successful but ...
- ✓ The server-side implementation needs more testing



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success
- ✓ Java server-side - Successful but ...
- ✓ The server-side implementation needs more testing
- ✓ Optimize applet code for server-side.



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success
- ✓ Java server-side - Successful but ...
- ✓ The server-side implementation needs more testing
- ✓ Optimize applet code for server-side.
- ✓ Overall Project - Big success



# Conclusions and Further research

- ✓ MISWEB Implementation - Not too good
- ✓ Java Applet - Success
- ✓ Java server-side - Successful but ...
- ✓ The server-side implementation needs more testing
- ✓ Optimize applet code for server-side.
- ✓ Overall Project - Big success
- ✓ DØ now has two systems for browsing the calibration database.



# Acknowledgement

Ghanaian Adinkra Symbols  
unity in diversity



Thank you SIST  
committee



# Acknowledgement

Ghanaian Adinkra Symbols  
wisdom, intelligence and patience



- ✓ Thank you SIST committee
- ✓ Thank you Taka Yasuda and Geoff Savage

# Acknowledgement

Ghanaian Adinkra Symbols  
friendship and interdependence



- ✓ Thank you SIST committee
- ✓ Thank you Taka Yasuda and Geoff Savage
- ✓ Thank you fellow interns

# The End

## The 5 Stages Of Debugging

At some point in each of our lives, we must face errors in our code. Debugging is a natural healing process to help us through these times. It is important to recognize these common stages and realize that debugging will eventually come to an end.



### Denial

This stage is often characterized by such phrases as "What? That's impossible," or "I know this is right." A strong sign of denial is recompiling without changing any code, "just in case."



### Bargaining/Self-Blame

Several programming errors are uncovered and the programmer feels stupid and guilty for having made them. Bargaining is common: "If I fix this, will you please compile?" Also, "I only have 14 errors to go!"



### Anger

Cryptic error messages send the programmer into a rage. This stage is accompanied by an hours-long and profanity-filled diatribe about the limitations of the language directed at whoever will listen.



### Depression

Following the outburst, the programmer becomes aware that hours have gone by unproductively and there is still no solution in sight. The programmer becomes listless. Posture often deteriorates.



### Acceptance

The programmer finally accepts the situation and goes to play some Quake.

This message brought to you by Oberlin College Computer Science 2002-03.

