H1OO Analysis Tools - Workshop

THE UNIVERSITY of LIVERPOOL

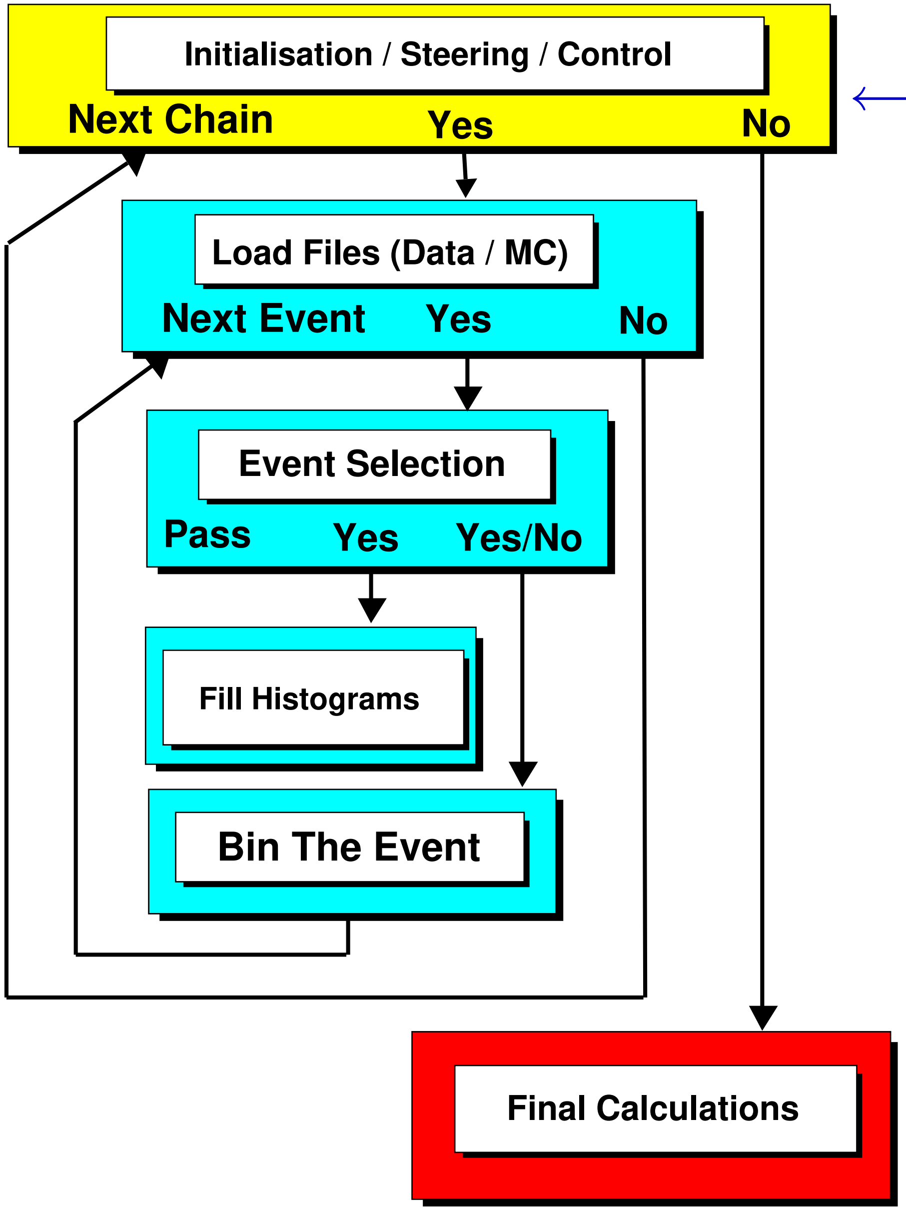
A GENERIC ANALYSIS FRAMEWORK

Paul Laycock and David South

Overview:

- Generalised Analysis Structure
- Generic Analysis Objects
- File Handling
- Variable Calculation and Systematics
- MC Studies
- An Example Analysis
- Summary

A Generalised Picture of Analysis



⟵ **Analysis Level**

Do something with Analysis Chains

⟵ **Analysis Chain Level**

Do something with Events

⟵ **Event Level**

- **Make Selections**
- **Fill Histograms**
- **Bin some quantities**

⟹ **User defines Event Loop**

⟹ **Structure is the same**

⟹ **Perfect for OO !**

GENERIC ANALYSIS OBJECTS - H1ANALYSIS

Analysis {abstract}

+ DoAnalysis(Analysis* a) : void ⟵ What to do with those **Analysis Chains**

+ GetBinGrid(TString name) : BinGrid*

+ GetSelector(TString name) : Selector*

 # AddBinGrid(BinGrid* bg) : void ⟵ Bin Grids and Event Selections

 # AddSelector(Selector* s) : void ⟵ Defined at Analysis Level

 # MakeChain(TString name) : void ⟵ User defines Chain Type

 # SetObjects() : void ⟵ User defines Object Attributes

- fChains : TObjArray* ⟵ Arbitrary number of

- fSelectors : TObjArray*

- fBinGrids : TObjArray*

- **Analysis Chains**
- **Selections**
- **Binning Schemes**

GENERIC ANALYSIS OBJECTS - H1ANALYSISCHAIN

AnalysisChain {abstract}

+ DoAnalysis(Analysis* a) : void
+ AddHistManager(HistManager* h) : void
+ AddMeasurement(Measurement* m) : void
+ GetHistManager(TString name) : HistManager*
+ GetMeasurement(TString name) : Measurement*

EventLoop(Analysis* a) : void

- fTree : H1Tree*
- fHistManagers : TObjArray*
- fMeasurements : TObjArray*

← **What to do with those Events**

Defaults for Data,Background And Signal Chain Types

← **User defines Event Loop**

← **Arbitrary number of**

- **Hist Managers**
- **Binned Measurements**

Bin Grids and Event Selections defined at Analysis Level

User defines how to use them to produce Histograms and Measurements

Histogram Management

HistManager {abstract}

+ Fill(int index) : void
+ GetIndex(TString name) : int

BookHistos() : void
FillHistos(int index) : void
cd(int index) : void

- fHistArrays : TObjArray*
- fCurrentHistArray : TObjArray*

What to do with those Histograms

User defines Histograms

Filling Code

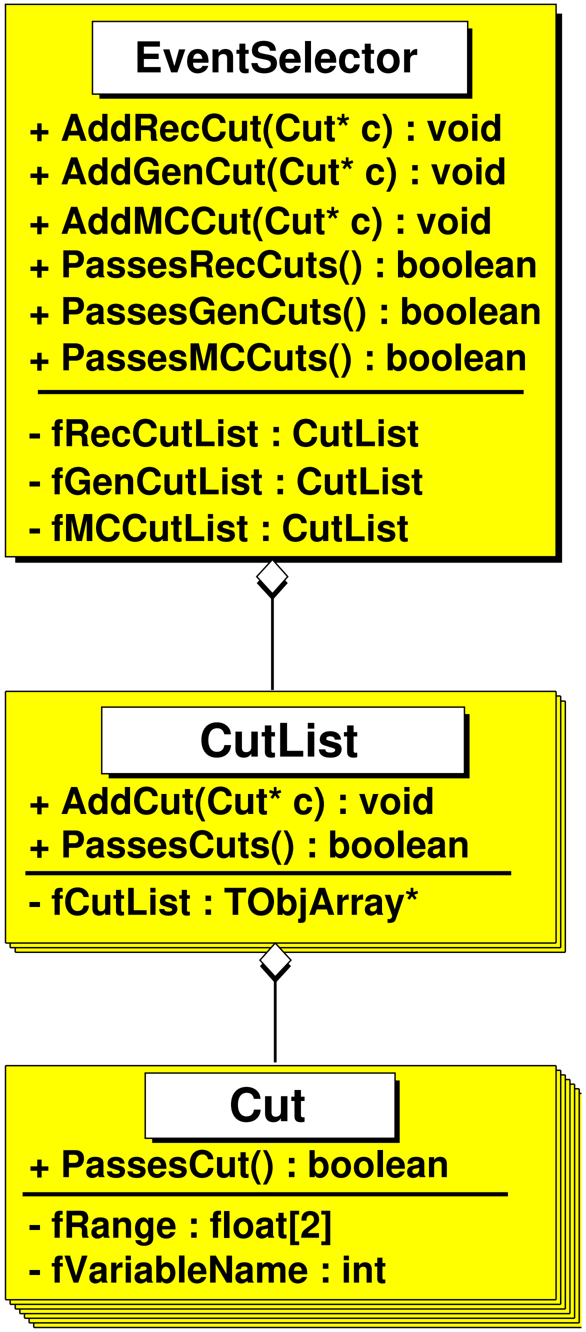
Arbitrary number of

- Arrays of Histograms

Utility functions for booking and filling

Plotter class used to produce control plots

EVENT SELECTIONS



⟵ What to do with those Cuts

Define three levels of event selection

- Reconstructed level
- Generator level
- MC phase space level

⟶ User defines Cuts

⟶ Experts define selections

⟶ Experts define selections for other users

⟶ Experts also define relevant hist managers

⟶ **Easy proliferation of expert knowledge**

File Handling

Use recently updated H1SteerTree to easily load HAT, mODS, ODS, User trees

Load one set of files per Analysis Chain, each H1SteerTree has a name

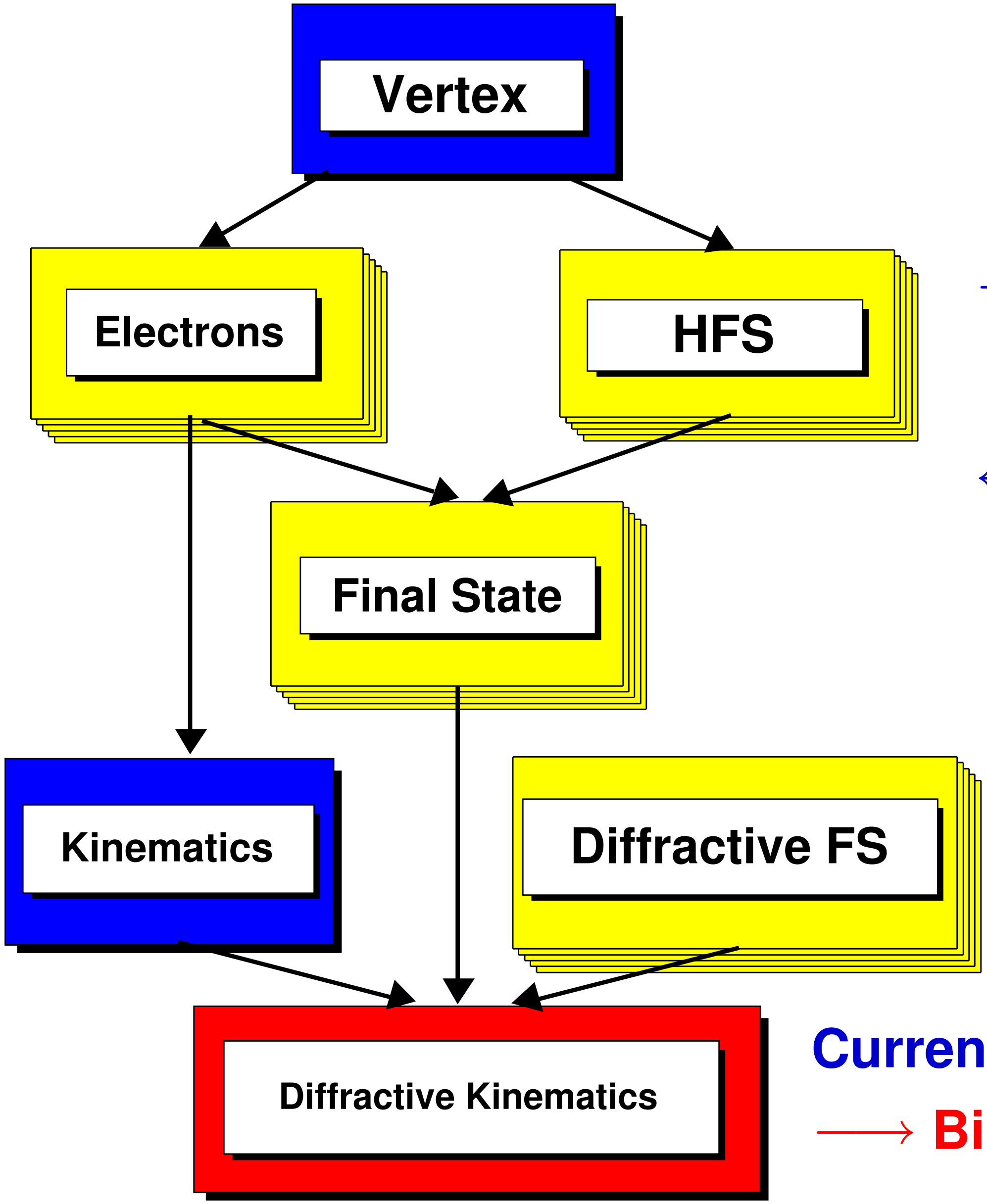
Easily steer which files for which Analysis Chain - name is a string in a text file

H1SteerTree stores:

- Phase space parameters for mixing files
- The luminosity
- Generated Zvtx parameters

Weight calculated in H1CalcWeight

Variable Calculation and Systematics



$\longleftarrow$ What to do with those

Very many variables

$\longrightarrow$ Systematics requires consistency

$\longrightarrow$ Singleton Event Object

$\longleftarrow$ Currently use H1Calculator

Provides generic interface

Event::GetVariable(int ID)

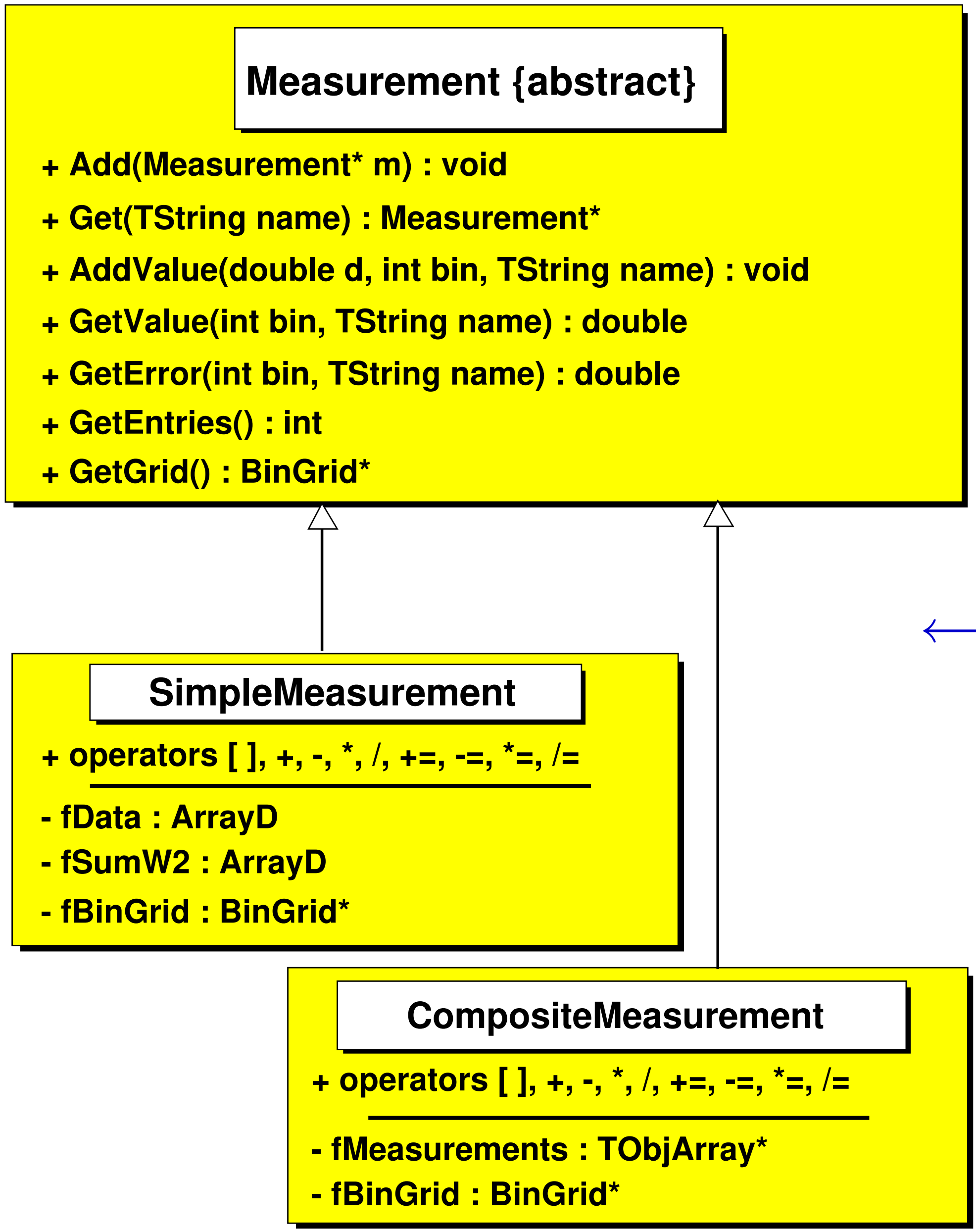
Other objects only need enum ID

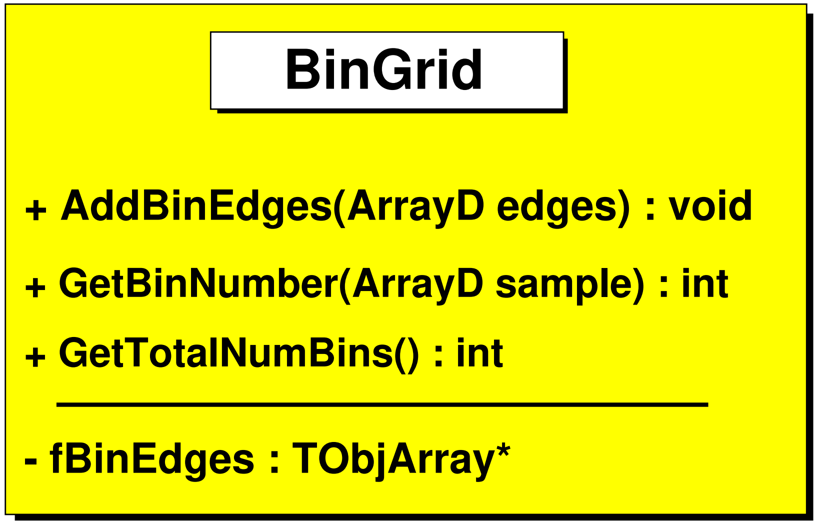
Event::GetVariable(Kine_Q2)

Current H1Calculator has 473 variables

$\longrightarrow$ Big and getting bigger, needs thought

MC Studies 1





← **What to do with those**

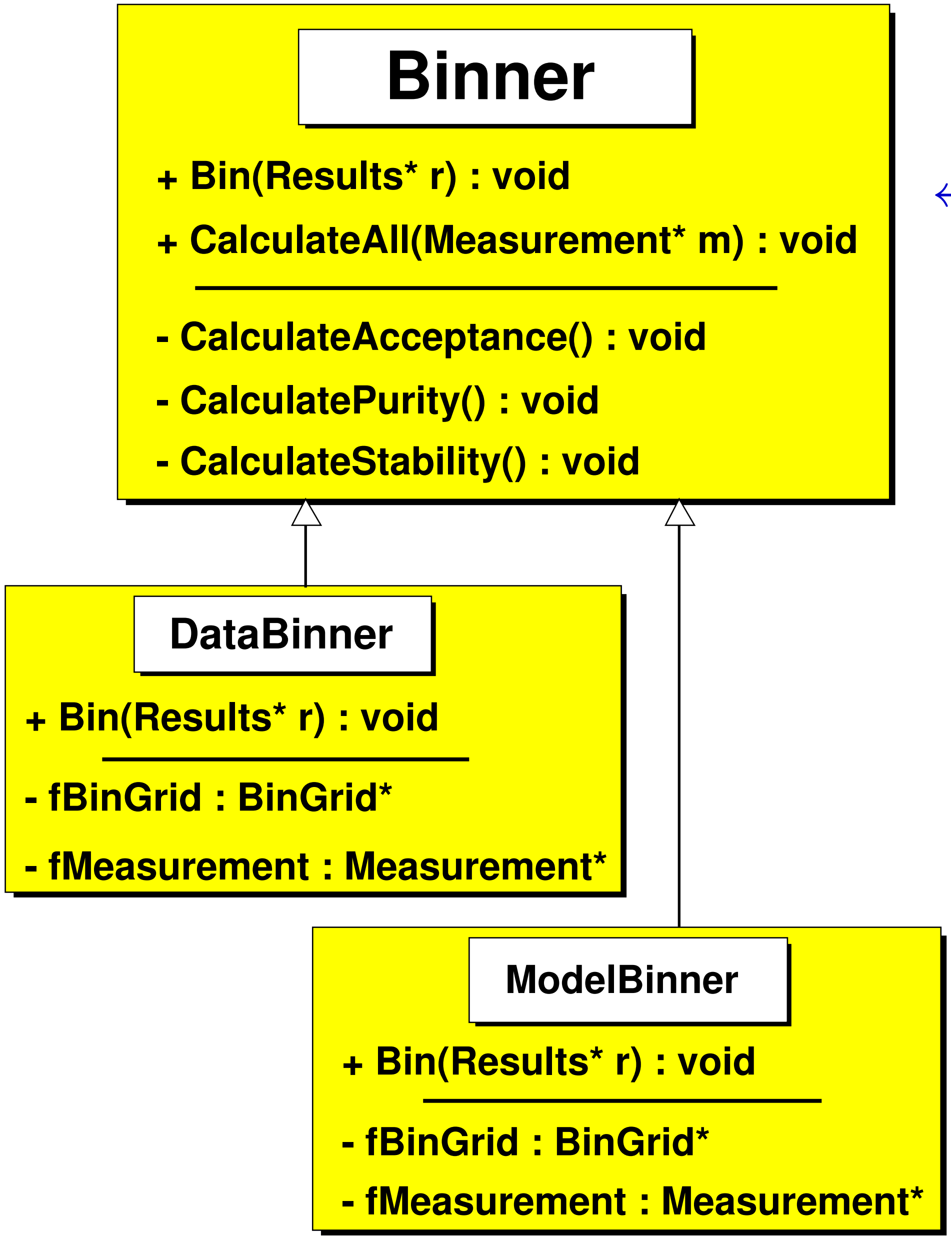
Very important numbers !

→ **User defines Bin Grid(s)**

→ **N dimensions**

→ **Binned variables steerable**

MC Studies 2



⟵ **How to calculate those**

Very important numbers !

⟶ **User defines Binner name !**

⟶ **Choose Data or Model**

Binner calculates:

- **Acceptance**
- **Purity**
- **Stability**

⟶ **using binomial errors**

Summary

Many common tasks to be done in analysis

→ Use OO to take advantage and generalise

- **Common framework/structure possible**
  - **Quick startup time for new students**
  - **Quick startup time for physicists (no offence to students !)**
- **Store the entire Analysis object on file**
  - $\Longrightarrow$ **Know** *exactly* **what was done in that analysis**
- **Quick and easy proliferation of expert analysis knowledge**
  - $\Longrightarrow$ **Plug in expert selections and histogram managers**

$\Longrightarrow$ Faster turn around for H1 publications

$\Longrightarrow$ Now David will explain the ExampleAnalysis package ...