Fermilab

Unidirectional Build Architecture: Refactoring a HEP Data Analysis Codebase

Dheeran Wiggins[1,2], Dr. Peter Dong[1,2]

[1]*Illinois Mathematics and Science Academy*, [2]*Fermilab*

On behalf of the CMS Collaboration

IMSA

imsa.edu

Preface

- Doubly charged Higgs boson ($H^{\pm\pm}$) is a theoretical non-Standard Model scalar boson produced via Drell-Yan mechanism

- Dark photon (γ_d) is a theoretical dark matter non-SM gauge boson in the dark sector produced via Higgs and SUSY portals

Preface

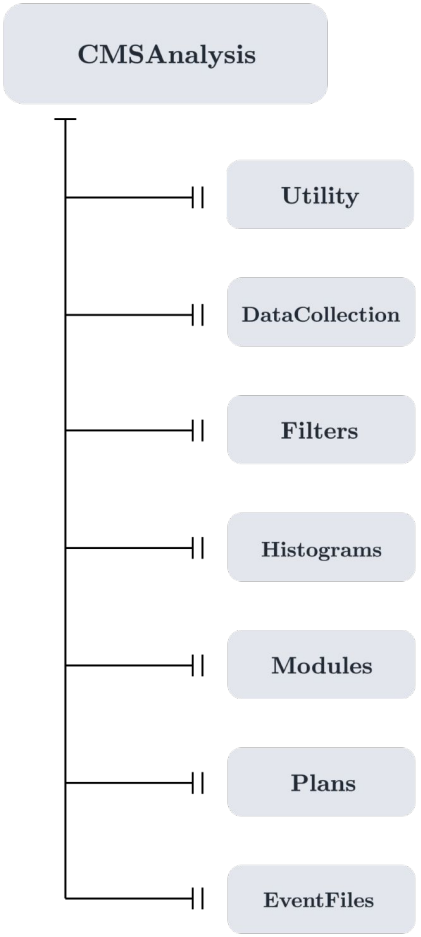
- Using collected tracking and calorimeter data from the CMS experiment, alongside simulated PYTHIA models, *pp* collision event data is sifted through:
  - statistical methods
  - identification algorithms
  - filters

Motivation

- All event data monitoring code (~300 files) is stored within DataCollection
- A single directory for such diverse tasks proved increasingly inefficient in:
  - code management
  - compilation times
- New codebase architecture for the $H^{\pm\pm}$ and γ_d searches must be explored

Proposed Architecture

- Split DataCollection into 7 task-oriented directories
- Selectors, filters, triggers, and cuts under Filters
- Efficiencies and modules assigned to Modules
- Widely used files under Utility

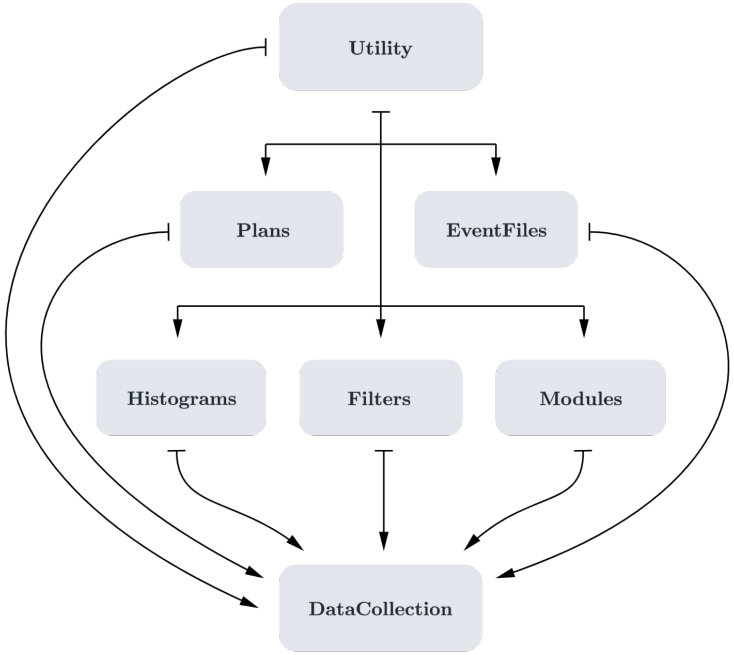


Refactoring Goals

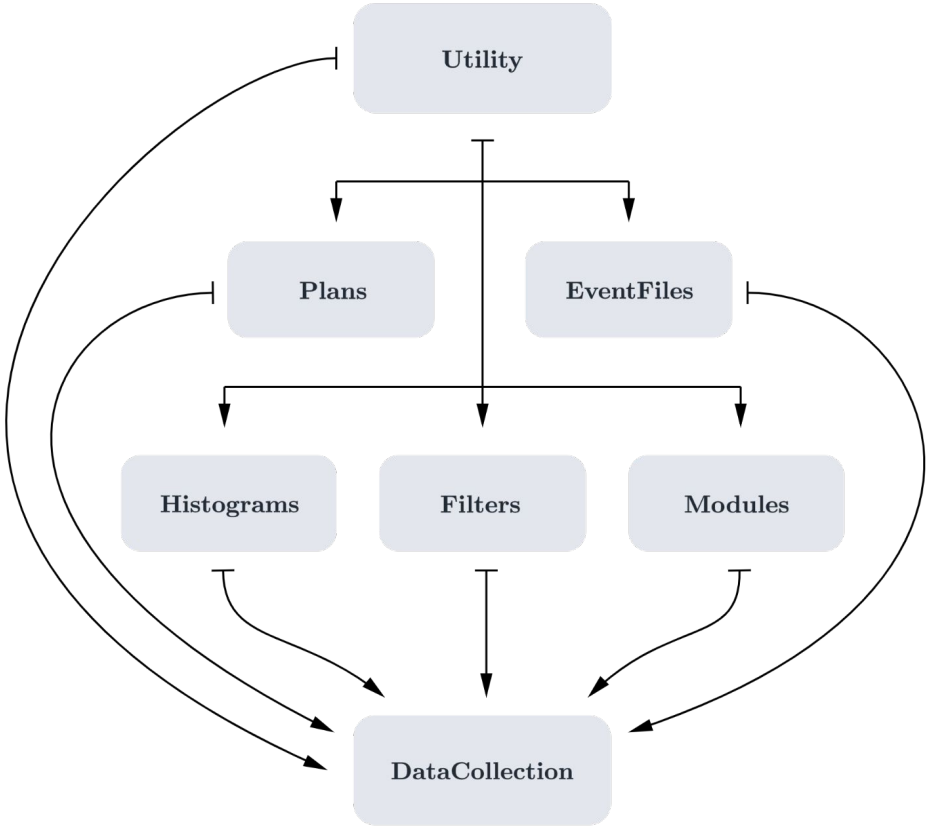
- Improve organization of code
- Alter classes that do not fit within revised architecture
- Restructure dependencies
- Maintain functionality of all classes
  - internal changes do not affect the files' external behavior

Directives Hierarchy

- Initial sources must be higher on hierarchy
- Terminal files lower



Directives Hierarchy



Refactoring Methods

- Each BuildFile.xml configuration file for the subdirectories scripted to determine build targets per the designed hierarchy
- Avoid circular dependencies!
- Include directives rewritten across the entire set of subdirectories
- Reconstructed problematic classes in new files to fit architecture and keep functionality

Summary

- Files within DataCollection were sorted into one of seven categories
- Categories based on relative purpose in the directory, with like-task files grouped together
- Utility as an initial source for all and terminal for none and DataCollection as terminal for all and initial for none