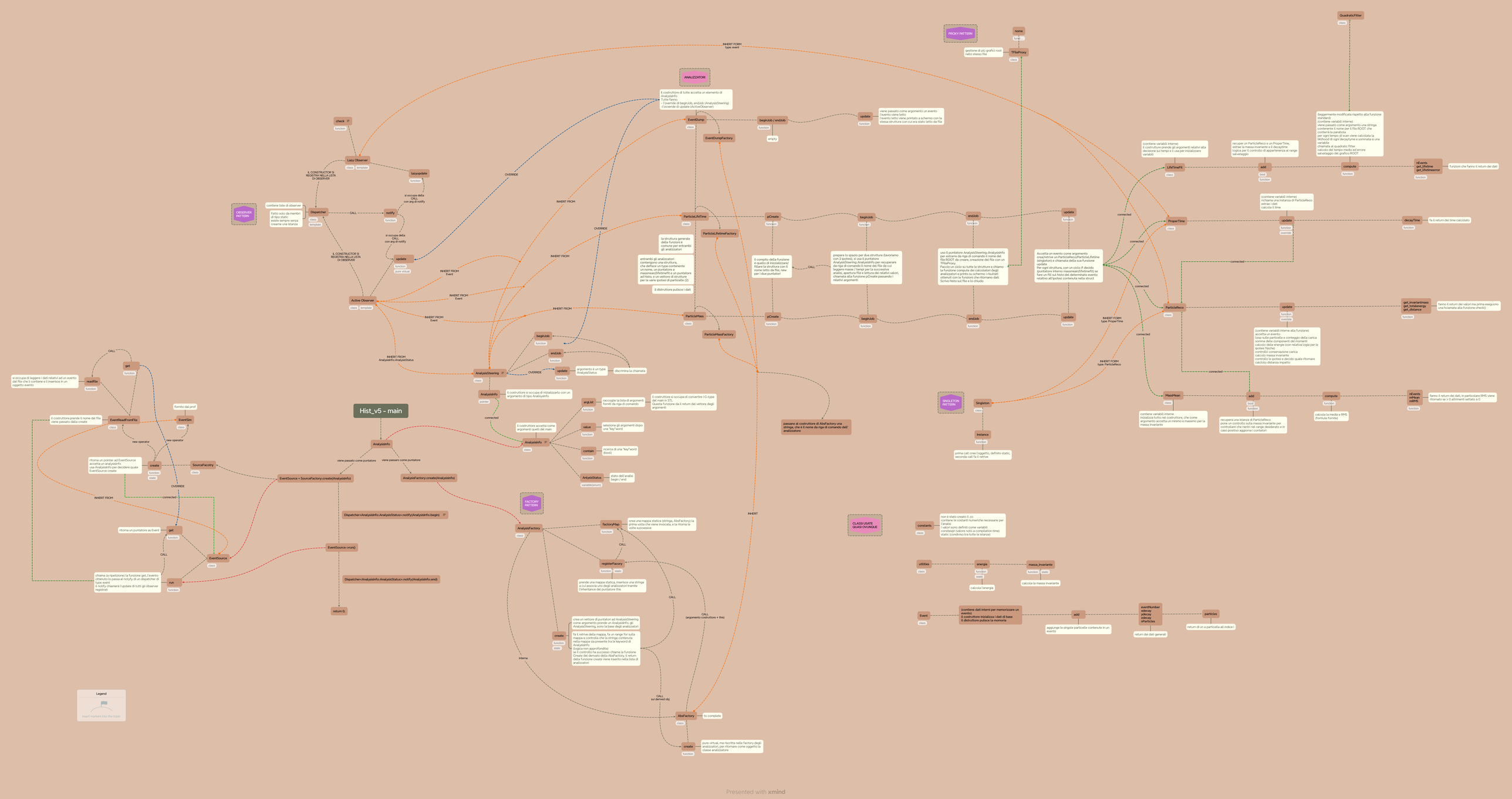
Hist_v5 - main
OBSERVER PATTERN
Lazy Observer
class template
lazyupdate
function
si occupa della CALL con arg di notify
check
function
IL CONSTRUCTOR SI REGISTRA NELLA LISTA DI OBSERVER
Dispatcher
class template
contiene lista di observer
Fatto solo da membri di tipo static
esiste sempre senza crearne una istanza
CALL
notify
function
si occupa della CALL con arg di notify
update
function pure virtual
Active Observer
class template
INHERIT FROM Event
INHERIT FROM Event
INHERIT FROM Event
INHERIT FROM AnalysisInfo,AnalysisStatus
OVERRIDE
INHERIT FROM
OVERRIDE
INHERIT FROM
ANALIZZATORI
Il costruttore di tutte accetta un elemento di AnalysisInfo
Tutte fanno:
- l'override di begin/job, endJob (AnalysisSteering)
- l'override di update (ActiveObserver)
EventDump
class
EventDumpFactory
begin/job / endJob
function
empty
update
function
viene passato come argomento un evento
l'evento viene letto
l'evento letto viene printato a schermo con la stessa struttura con cui era stato letto da file
ParticleLifeTime
class
ParticleLifeTimeFactory
la struttura generale (vedi Fundamental) comune per entrambi gli analizzatori
entrambi gli analizzatori contengono una struttura, che definisce un type contenente un nome, un puntatore a massmean,lifetimeFit e un puntatore ad histo; e un vettore di strutture per le varie ipotesi di particelle (2)
il distruttore pulisce i dati
pCreate
function
il compito della funzione è quello di inizializzare/ fillare la struttura con il nome letto da file, nei per i due puntatori
CALL
begin/job
function
prepara lo spazio per due strutture (lavoriamo con 2 particelle) e chiama l'operatore AnalysisSteering::AnalysisInfo per recuperare da riga di comando il nome del file da cui leggere massa / tempi per la successiva analisi, apertura file e lettura dei relativi valori, chiamata alla funzione pCreate passando i relativi argomenti
endJob
function
uso il puntatore AnalysisSteering::AnalysisInfo per estrarre da riga di comando il nome del file ROOT da creare, creazione del file con un TFileProxy
Faccio un ciclo su tutte le strutture e chiamo la funzione compute dei calcolatori degli analizzatori e printo su schermo i risultati ottenuti con le funzioni che ritornano dati. Scrivo histo sul file e lo chiudo
update
function
Accetta un evento come argomento crea/retrive un ParticleRecoParticleLifetime (singlaton) e chiamata della sua funzione update
Per ogni struttura, con un ciclo if decido (questo è interno mesmean,lifetimefit) se fare un fill sul histo del determinato evento relativo all'ipotesi contenuta nella struct
ParticleMass
class
ParticleMassFactory
pCreate
function
begin/job
function
endJob
function
update
function
INHERIT FROM type ProperTime
INHERIT FROM type ParticleReco
AnalysisSteering
class
OVERRIDE
begin/job
function
endJob
function
update
function
argomento è un type AnalysisStatus
discrimina la chiamata
AnalysisInfo
pointer
Il costruttore si occupa di inizializzarlo con un argomento di tipo AnalysisInfo
connected
AnalysisInfo
class
Il costruttore accetta come argomenti quelli del main
argList
function
raccoglie la lista di argomenti forniti da riga di comando
Il costruttore si occupa di convertire i C-type dal main in STL
Questa funzione da il return del vettore degli argomenti
value
function
seleziona gli argomenti dopo una "key"word
contain
function
ricerca di una "key"word (bool)
AnalysisStatus
variabileenum
stato dell'analisi begin / end
passano al costruttore di AbsFactory una stringa, che è il nome da riga di comando dell' analizzatore
INHERIT
PICKY PATTERN
nome
function
TFileProxy
class
gestione di più grafici root nello stesso file
QuadraticFitter
class
(contiene variabili interne)
il costruttore prende gli argomenti relativi alla decisione sul tempo e li usa per inizializzare variabili
LifeTimeFit
class
recuper un ParticleReco e un ProperTime, estrae la massa invariante e il decaytime
logica per il controllo di appartenenza al range salvataggio
add
tool
function
Eseguermente modificata rispetto alla funzione standard)
(contiene variabili interne)
viene passato come argomento una stringa contenente il nome per il file ROOT che contiene la parabola
Per ogni evento dicaun viene calcolata la likihood di ogni decaytime e sommata il una variabile
chiamata al quadratic fitter
calcolo del tempo medio ed errore salvataggio del grafico ROOT
compute
function
nEvents
get_lifetime
get_lifetimeerror
function
funzioni che fanno il return dei dati
ProperTime
class
(contiene variabili interne)
richiama una instanza di ParticleReco
estrae i dati
calcola il time
update
function
override
decayTime
function
fa il return del time calcolato
ParticleReco
class
update
function
override
get_invariantmass
get_totalenergy
get_distance
function
fanno il return dei valori ma prima eseguono una hcianata alla funzione check()
(contiene variabili interne alla funzione)
accetta un evento
loop sulla particelle e contteggio della carica
calcolo delle componenti del momenti
calcolo della energie (con relativa logia per le ipotesi fisiche)
controllo conservazione carica
calcolo massa invariante
controllo le ipotesi e decido quale ritornare
calcolo distanza impatto
MassMean
class
contiene variabili interne
inizializza tutto nel costruttore, che come argomento accetta un minimo e massimo per la massa invariante
add
tool
function
recupera una istanza di ParticleReco
estrae la massa invariante variabile per controllare che rientri nel range desiderato e in caso positivo aggiorna i contatori
compute
function
calcola la media e RMS (formula furnsta)
nEvents
mMean
mRMS
function
fanno il return dei dati, in particolare RMS viene ritornato se > 0 altrimenti settato a 0
SINGLETON PATTERN
Singleton
class
Instance
function
prima call crea l'oggetto, definito static, seconda call fa il retrive
readFile
function
CALL
get
function
si occupa di leggere i dati relativi ad un evento dal file che li contiene e li inserisce in un oggetto evento
EventReadFromFile
class
Il costruttore prende il nome del file viene passato dalla create
fornito dal prof
EventSim
class
new operator
new operator
create
function
static
ritorna un pointer ad EventSource
accetta un analysisInfo
usa AnalysisInfo per decidere quale EventSource create
SourceFacotry
class
OVERRIDE
connected
INHERIT FROM
ritorna un puntatore ad Event
get
function
CALL
EventSource
class
run
function
chiama (a ripetizione) la funzione get, l'evento ottenuto passa al notify di un dispatcher di type event
il notify chiamerà l'update di tutti gli observer registrati
EventSource = SourceFactory::create(AnalysisInfo)
AnalysisFactory::create(AnalysisInfo)
viene passato come puntatore
viene passaro come puntatore
Dispatcher<AnalysisInfo-AnalysisStatus>-notify(AnalysisInfo-begin)
EventSource->run()
Dispatcher<AnalysisInfo-AnalysisStatus>-notify(AnalysisInfo-end)
return 0;
FACTORY PATTERN
AnalysisFactory
class
factoryMap
function
crea una mappa statica (stringa, AbsFactory) la prima volta che viene invocata, e la ritorna la volte successive
CALL
registerFacory
function
static
prende una mappa statica, inserisce una stringa a cui associa uno degli analizzatori tramite l'inheritance del puntatore this
CALL
CALL
(argomento costruttore + this)
create
function
static
crea un vettore di puntatori ad AnalysisSteering come argomento prende un AnalysisInfo, gli AnalysisSteering, sono la base degli analizzatori
fa il retrive della mappa, fa un range for sulla mappa e controlla che la stringa contenuta nella mappa sia presente tra le keyword di AnalysisInfo
Foglia non approfondita
se il controllo ha successo chiama la funzione Create del derivato della AbsFactory, il return della funzione create viene inserito nella lista di analizzatori
interna
CALL
sui derived obj
AbsFactory
class
to complete
create
function
pure virtual, ma riscritta nelle factory degli analizzatori, per ritornare come oggetto la classe analizzatore
CLASSI USATE QUASI OVUNQUE
constants
class
non è stato creato il .cc
contiene le costanti numeriche necessarie per l'analisi
I valori sono definiti come variabili constexpr (valore noto a compilation time)
static (condiviso tra tutte le istanze)
utilities
class
energia
function
static
calcola l'energia
massa_invariante
function
static
calcola la massa invariante
Event
class
(contiene dati interni per memorizzare un evento)
Il costruttore inizializza i dati di base
Il distruttore pulisce la memoria
add
aggiunge le singole particelle contenute in un evento
eventNumber
xdecay
ydecay
zdecay
nParticles
return dei dati generali
particles
return di un a particella all'indice i
Legend
Insert markers into the topic
Presented with xmind