Tracking Event Model, Status

Status of the Tracking Event Model

Jose A. Hernando, E. Rodrigues

Visiting the Plan

Future work:

Before 3/05, and later…

Visiting the planning

Step 1: **Track, State, IExtrapolator**

- **Goal: standard output of the Fitting Algorithms (on/off)line**
- **Steps:**
  - ***Agreement in the base classes***
  - ***Implementation of converters*** ←
  - ***Modify client/tracking code to use these classes***

Step 2: **Measurement, Node, IProjector**

- **Goal: common base classes for PR and FA algorithms**
- **Steps:**
  - ***Re-built all information from persistency***
  - ***Agreement in the base classes***
  - ***Use of Node, Measurement, Projector in the tracking code***

Agreement if the base classes

The classes: Track, State, Iextrapolator

1. Largely debated via e-mail and in presentations
2. A version with input from many people:
   1. We tried to combined different visions as much as possible
   2. But in some places we have to take a decision
3. Event Classes has two sides:
   1. persistency (optimized) + transient (in progress…)
4. It is time to use the classes and figure-out the problems

Step I: The clases, current view

Track

State

A TRACK:

flag (bitField) TYPE, HISTORY, FLAG

chi2/ndof, ndof (quality)

physics State = “The persistent State”

<LHCbID> = list of LHCbID

<States*> = “the *transient* states”

Methods:

Access to physics state: *p,pt, slopes, position*

Access states: *at z, plane, LOCATION*

A STATE:

flag (bitField) TYPE, LOCATION

vector-state, covariance, z

Methods:

Access to physics contents:

fix (x,y,tx,ty), overwritable q/p

TYPE: Velo, VeloR, Upstream, Dowstream, Ttrack, Long

HISTORY: Algorithm: ie TrgForward

FLAG: Valid,

TYPE: Linear, HasMomentum

LOCATION: BeginVelo, EndVelo, atTT,…

Step I: Converters and Client users

Step 1.2: Converters

- **Goal**: *Convert the output of the Fit Algorithms to Track/States*
- OnLine: TrFitTrack -> Track, TrgTrack <->Track
- **Status:** *Almost done, compile, need to check, done by end of the year*
  - *Problems: serializers, LHCbID muon, revisiting flags…*

Step 1.3: Use of the classes **(Track, State, IExtrapolator)**:

- **Goal:** *Client Algorithms (RICH, Muon) use Track/States*
- Clients: Replace and use Track/State/IExtrapolator
  - Please feed back us the problems…
- Tracking: How much you can use Track/State?, derived classes?
  - Trigger: what is left for *a TrgTrack, TrgState?*
- Others*: MC?, Tool to retrieve Clusters from LHCbID*
- **Status**: *I will say should be done by end of 2/05*

Step II: Recovering the track and more

Step 2.1: Recovering the track

- **Goal**: Recreate a Track from persistency, and refit it, from the list of LHCbID
- But: Is this possible?
- PR algorithms should be divided in two:
  - Standalone: return track segment(s) and a their LHCbID's
  - From a list of LHCbID: get a local track segment
    - We need to check this part!!
- **Status:** *will be nice to have a confirmation of this before end 2/05*

Step 2.2: Agreement of the internal tracking classes

- **Goal**: define the common classes (only for the tracking community)
- Measurement, Node, Projector

Step 2.3: Use the classes in the PR and FM algorithms

- **Goal:** To be able to share/add/remove easily PR and FM algorithms
- **Status:** *I see very, very unlike for 03/05, but this can wait*

Status and plans

Step I:

- Track/State/IExtraplator are OUT
    - “Multi”cultural classes but “No one” is perfect
- Converters:
    - FA tracks to Track, *almost done*

12/04

- Use of Track/State/Iextrapolator
    - Clients: go and used, tell us the problems
    - Tracking: replacing TrgTrack, TrFitTrack can wait

2/05

Step II:

- Recover and refit the track (from LHCbIDs)
    - Modify the PR algorithms

… 2/05

- Tracking internal classes: Measurement, Node, Iprojector
- Use of these classes

… /05