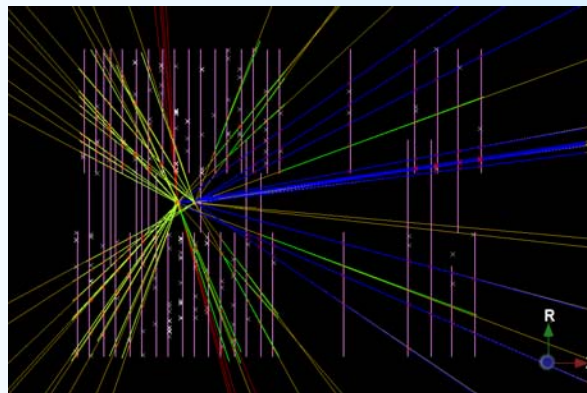


Tracking in LHCb

E. Rodrigues, NIKHEF



A Status Report

DC'06 Shopping list

On the shopping list

- ❖ Removal of all tracking code related to old event model
- ❖ Introduction of extra necessary tools & algorithms
- ❖ Introduction of LHCb namespaces
- ❖ Move from CLHEP to MathCore & SMatrix
- ❖ Adaptations to new detector elements
- ❖ Adaptations to realistic geometries -> introduction of trajectories
- ❖ Set-up of a new tracking sequence for Brunel

... last but not least: test and debug all of this!

“Technical” work

- **Removal from Brunel of all tracking code running with the old TEM**
 - ❖ *co-existence with our present TEM can be confusing*
 - ❖ *I still see such code (commented out) in Reco.opts ... can this be removed, please?*
 - ❖ *Also BrunelMC.opts, BrunelCheck.opts, DstContent.opts, MCDstContent.opts*
 - ❖ *Anywhere else?*

- **This required the introduction of new tools, algorithms and packages ...**

- ✓ **Tr/TrackMCInterfaces**
 - Collects all interfaces accessing MC information
 - ❖ *Just introduced in February*
 - ❖ *Decouples MC-related interfaces from interfaces needed for real data reconstruction*

- ✓ **Tr/TrackUtils**
 - Contains general tracking algorithms
 - ❖ *Contains only the event clone killer for now*
 - *Old algorithm replaced by “clone finder” tool + algorithm*
 - ❖ *Will probably evolve, e.g.:*
 - *Could be the place to add algorithm to prepare tracks for DST*
 - *E.g.: all but 1 state stored on the track by default -> need to “strip” tracks*

- ✓ **Tr/TrackSys**
 - Defines the global set of tracking packages
 - ❖ *Takes care of setting up the consistent set of tracking packages to use*
 - ❖ *Single place where the tracking sequence for Brunel can be set-up*

NEW

Note: there are also some tools introduced:

e.g.: TrackCaloMatch, TrackCloneFinder (both in TrackTools)

Contact us if you need some tool/functionality/... better check than double the work ...

LHCb namespaces, small Gaudi changes

- **Went rather smoothly, easy**

Move from CLHEP to MathCore and SMatrix

- **Not quite so smooth**
- **Tracking code required temporary hacks**
 - ❖ *now these are history*
- **Tracking also provided a lot of feedback and requested new functionality**
- **Note: some “features” are really hard to trace,
cf. the discussion on $A = A * B$ of last week ...**
- **We should be careful! The packages are rather new and may not be doing quite what we think or would like them to be doing ...**
 - ❖ *such “details” have required very many hours of debugging to be traced*

Towards a Realistic Tracking

Motivations and Goal

NEW

- **Tracking has to deal with realistic (= non-ideal) geometries**
- **Track fitting should be as blind as possible to sub-detector details**
- **Introduce Trajectories à la BaBar to “solve everything”**

Needs

- **Trajectory-like event model classes**
- **Connection tracking – (tracking) detector elements**
 - ❖ *detector shapes communicated to tracking via trajectories*
 - > *the tracking can then handle arbitrary shapes/implementations*
- **Adaptation of all track fitting code to new detector elements, trajectories**

Note:

see Edwin's presentation @ the Lausanne tracking workshop for all technical details

Classes & tools developed

- **Trajectory classes for strips / wires**
 - Trajectory base class & DifTraj template class
 - LineTraj, ParabolaTraj and CircleTraj
- **Trajectory class for a State**
 - StateTraj
- **Tool to calculate the Point Of Closest Approach between two Trajectories**
 - TrajPoca with ITrajPoca interface
- **Projector tools based on Trajectories**
 - TrajOTProjector, TrajSTProjector, TrajVeloRProjector and TrajVeloPhiProjector
- **Subdetector people helped making the detector elements provide Trajectories**

NEW

Intermezzo:
“Playing” in Python
with the Tracking

A lot of effort has been put into getting
all the necessary
to use in Python the tracking classes and tools

Dictionaries

- All our XML-defined Event classes have the corresponding dictionaries built
- Other event classes defined in .h & .cpp needed some extra “hand-made” files for producing the dictionaries

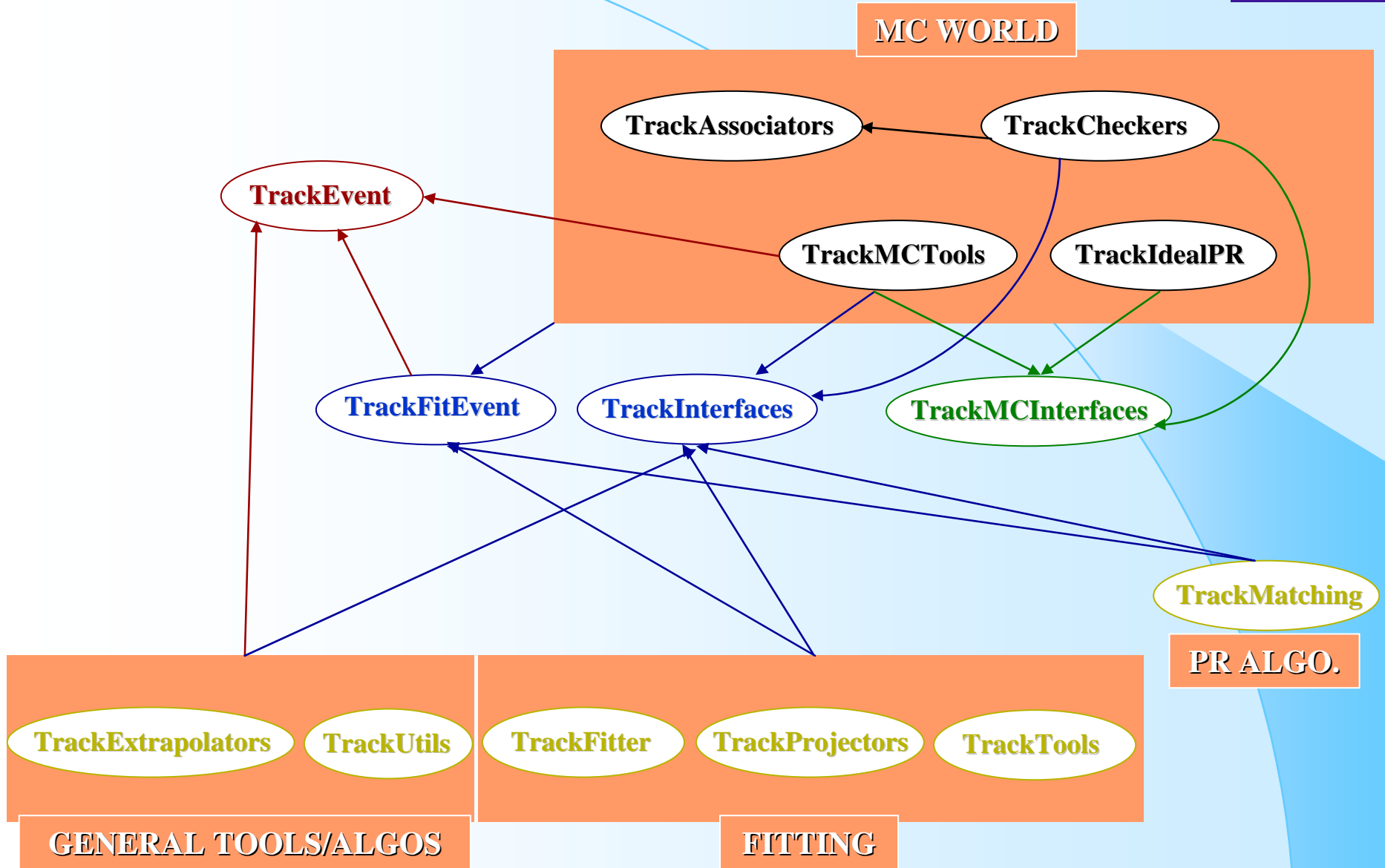
Useful?

- **Enormously!**
- **I've used the framework extensively for testing and debugging ...**
- **Do not hesitate to get into it ...**
- **A note on *Tracking in Python* is about to be released ...**

Overview of the Tracking Packages

| Event Model | General Tools | Miscellaneous |
|---|--|--|
| <p>Event/TrackEvent Kernel/LHCbKernel</p> <p>Pattern Recognition</p> | <p>Tr/TrackInterfaces Tr/TrackExtrapolators Tr/TrackTools Tr/TrackMCInterfaces Tr/TrackMCTools</p> <p>Fitting</p> | <p>Tr/TrackUtils Tr/TrackSys Tr/TrackPython</p> <p>MC Association</p> |
| <p>Tr/TrackMatching</p> | <p>Tr/TrackFitEvent Tr/TrackProjectors Tr/TrackFitter</p> | <p>Tr/TrackAssociators Tr/TrackCheckers</p> <p>Tr/TrackIdealPR</p> |

Packages Dependencies



Status of the Tracking

Testing & Debugging

A lot of bugs found

- ❖ *Introduced when moving to new detector elements*
- ❖ *Introduced when adapting to MathCore and SMatrix*
- ❖ *Introduced when going into the “trajectories world”*
 - *Surprisingly few, as a matter of fact!*
 - *Most changes were improvements and additions*
- ❖ *Some nasty bugs were introduced by non-tracking people ...*
 - *When you edit tracking code, please take more than 15 seconds ...*
 - *... it may save the others hours of debugging ...*

Tracking Packages

- ✓ **Event model and interfaces rather stable**
- ✓ **General tools and algorithms also stable and seem doing the right job**
- **Fitting part (TrackFitter, TrackProjectors) requires active testing and debugging**

TrackSys v2r0 contains full consistent list of TrackXxx packages

On the good side ...

Fitting tracks obtained with a cheated pattern recognition

- *Success rate typically ~99.9% for long tracks with loose quality cuts*
- *Some encouraging plots have been produced for the first time with our brand new DC'06 framework ... see next slide*
- *Still, quality plots shown there is more to it ... to be continued ...*

But

Fitting reconstructed long tracks produced by PatForward

- *Success rate goes down to ~75% ... why?*
- *Too early to say: could be a bug in PatForward, as a few days ago ... (crazy seed states)*
- *In any case there is one difference between cheated and rec. long tracks: both are long tracks but the ones from PatForward contain no ambiguity info!*

Bottom line

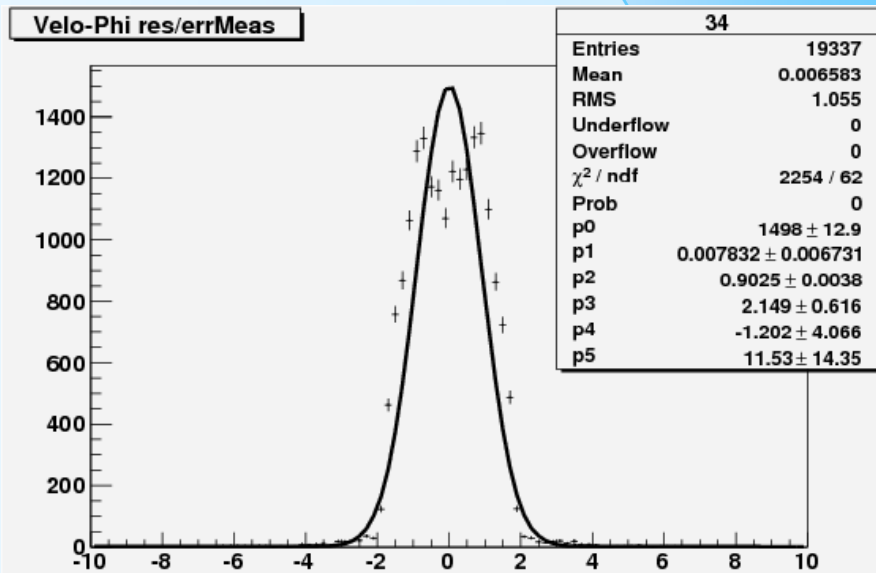
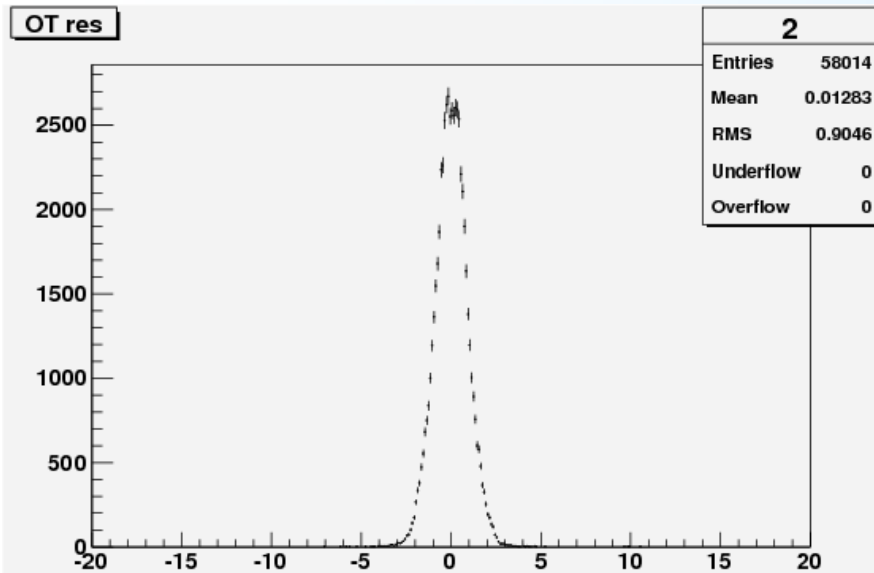
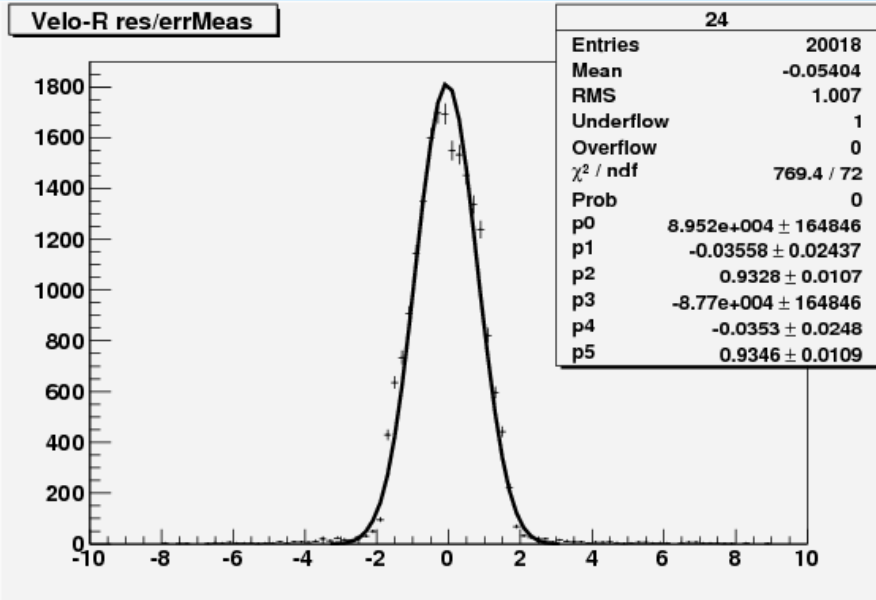
Is the issue of the OT ambiguities hitting us now badly?

- *An algorithm to determine ambiguities is on the shopping list ...?*

The fitting code does not crash anymore but needs more testing

First results on fitting
with DC'06 software!

All using the cheated
pattern recognition



Next Steps Towards DC'06

Fitting in Brunel

- The code does not seem to crash anymore
- Could be included in Brunel, for experts
- Do not expect amazing results right now with reconstructed long tracks

Track Fitting Tests

- Testing largely ongoing
- Tricky since many possible sources of problems

**In any case the results are impressive given that
« everything » has changed and the tracking is
working with version v0r* of the trajectories
framework!**

Tracking sequence for Brunel

Brunel v30r1 in LHCbDEV is rather incomplete

- *Will get quite some updates from the tracking in the next days ...*

Seeding, Matching, Velo-TT and clone killing algorithms

- *To be included in next Brunel*
- *Will use for the moment a cheated seeding*
 - *Needed to start tuning the Matching – not touched/looked at/tuned for over a year ...*

Tracking options for Brunel

- *Trying to have one options file for the tracking*
 - ✓ *Clear and simple*
 - ✓ *Easy to check and change/adapt for dedicated studies*
- *Reco.opts would simply become*

```
// Tracking reconstruction phase: pattern recognition + fitting
#include "$TRACKSYSROOT/options/RecoTracking.opts"
```

From TrackSys package

New Algorithms and Tools

Tool to prepare/pack the tracks to be stored on DST

- *Clone killer algorithm provides “best” set of tracks for physics*
- *These tracks still have too much info that is not to be stored on DST*
- *Need a procedure to clean these tracks: all but first state, all measurements, etc.*

Track filters

- *Alignment developers have already raised the question of how to easily get e.g. a long track with only VELO / seed hits (LHCbIDs)*
- *Seems like a “stripping” tool could be handy ...*
- *We have already some very simple track selector tool, but one could make a set of filter tools à la DaVinci (FilterCriterion family of tools)*

Tracking Monitoring

- *Some first thoughts thrown at the Lausanne workshop*
- *Work will start when possible, likely during DC'06 ...*

Documentation

- **Understood that the TEM & related needs to be documented ...**
- **Note on *Tracking in Python* about to be released**
- **Note on *Track Event Model* started to be written**
- **Tracking-related webpages are available:**
 - ❖ *Wiki page <https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbTrackFitting>*
 - ❖ *<http://lhcb-reconstruction.web.cern.ch/lhcb-reconstruction/>*
- **Tracking experts are welcome to go and add information / doc / HowTo's ... to the track fitting wiki page ...**