

Table of Contents

A-Team Meetings.....	1
Meeting Wednesday May 2nd 2007, 1.30 pm.....	1
Minutes.....	1
Meeting Tuesday April 3 2007, 10am.....	1
Agenda.....	1
Minutes.....	1

A-Team Meetings

Meeting Wednesday May 2nd 2007, 1.30 pm

To join the meeting, either go to <https://audioconf.cern.ch/call/0178298> or dial in:

- Call: +41227676000 (Main)
- Enter access code: 0178298

Minutes

- Kurt has started to write a set of small scripts to generate our private nightly and run standard test jobs. We need to find out which machines are available, if we can get a separate account, what data to run on etc.
- Wouter has used valgrind to make call trees of some parts of the pattern recognition. This may help in figuring out what exactly is slow, if anything.
- We have agreed that Stephanie, Kurt and Wouter are all at cern for extended periods starting May 29th. Chris will join approximately one week later.
- Next meeting: Wednesday May 23, 1.30pm

Meeting Tuesday April 3 2007, 10am

To join the meeting, either go to <https://audioconf1.cern.ch/call/0114967> or dial in:

- Call: +41227676000 (Main)
- Enter access code: 0114967

Agenda

- Pat overview (Stephanie) [↗](#)
- Tsa overview (Wouter) [↗](#)
- Data containers (Chris)
- Geometry regions (Kurt)
- Organizational issues

Minutes

First overview of Pat and Tsa needs. Some relevant points, not in links above

- *Trajectories*: Both Pat and Tsa access geometry via Trajectories and both re-interpret the Trajectory as a line. Pat caches the information in regions and performs its internal calculation of strip/wire position from the cached information. Tsa uses the Trajectory for each hit, but creates its own Tsa::Line object from the Trajectory. One solution would be to give the Detector Elements an interface to a cache resembling the Pat solution. What we need to figure out is:
 - ◆ Do we need the Trajectories? Once those are in a memory pool, do we really save time not using them?
 - ◆ Can we make Detector Elements return Trajectories of an explicit type, such that we can provide an inline interface to what the PR need to know? (This basically comes down to *not* exploiting the common base of CircleTraj (VeloR) and LineTraj (everything else).)
- *Decoding on demand*: To speed up data access in the trigger, we need to provide the functionality to decode the data for certain geometry regions of the detector on demand. This (probably) means
 - ◆ The PR algorithms access all event data via a tool, not directly from the TES

ATeamMeetingCoordinates < LHCb < TWiki

- ◆ The 'geometry' granularity should probably be a 'region'. The 'decoding' granularity is a readout buffer. Therefore, somewhere there must be a map from Detector Elements to buffers (not necessarily inside our framework). We need to specify an interface for the decoding tool.
- Exact knowledge of dead regions might become an issue. In the velo everything known is in a database and the interface exists to access it.

Task list

- Kurt: think about package structure, naming. Setup a test release.
- Stephanie: look precisely at structure of hits in both packages. Think about what our hit class(es) should look like to accommodate both Pat and Tsa.
- Chris: compare different container solution in a simple test program. Find out what it takes to obtain PatDataStore performance.
- Wouter: try to identify timing issues in decoding.
- All: contribute to sketch of data and interface classes

Next meeting: Wednesday May 2nd, 14.00.

-- WouterHulsbergen - 02 Apr 2007

This topic: LHCb > ATeamMeetingCoordinates

Topic revision: r9 - 2007-05-22 - WouterHulsbergen



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback