

Table of Contents

Geometry Regions and Geometry-aware Coordinates.....	1
Introduction.....	1
Geometry Regions.....	1
What is this?.....	1
Where they should be implemented.....	1
Geometry-aware Coordinates.....	2
What is this?.....	2
How and where to implement them.....	2

Geometry Regions and Geometry-aware Coordinates

Introduction

The purpose of this page is to share some ideas about geometry regions and geometry-aware coordinates. Currently this is just a white board for discussions. Eventually it will document our design decisions.

Geometry Regions

What is this?

First let's make sure we know what we are talking about. Personally I would prefer to use the term "geometry meta-structures". However, the term "region" was established by Olivier C. and he uses it throughout the Pat implementation and in his recommendations document. So let's not introduce too much new jargon.

A geometry region is a higher level structure of the geometry that does not necessarily have a direct representation in the geometry description. For example the Velo geometry description has direct representations of sensors and a Velo half. But no representation of a Velo module. Both, the Velo half and the module qualify as a geometry region. In addition, a geometry region can be something more granular than the smallest element of the geometry description. The four sectors on Velo R-sensors are an example. For our purpose geometry regions are semantic structures facilitating the implementation of pattern recognition algorithms. One can think of many such regions. The ones that are useful are most easily identified by describing a pattern recognition algorithm. Any geometrical concept needed to specify an algorithm has to be implemented by the software. In my opinion this implies it should be implemented **explicitly**.

Where they should be implemented

The software geometry representation at LHCb has two major components:

- the geometry (XML-)database
- the Detector Element interfaces

In the geometry database the detector components are represented in the usual way as a tree of nodes with relative positions and orientations. Details of this are not relevant to us. We can treat this as a black box. It is simply the source used to initialize the Detector Element interface classes.

The purpose of Detector Element interfaces is to grant access to the underlying geometry description. They also handle conditions that change with time, e.g. alignment. And now here it comes:

The Detector Element packages are the right, and the only right, place to implement any concept of geometry region.

Note that I say packages not classes. Of course one can introduce all kinds of little helpers that are not technically Detector Elements.

Currently this is not the case. For the Velo many region concepts are implemented in Pat in a non-transparent way. There are also implicit uses of geometry regions. For example, the Velo pattern recognition code assumes that the sensor numbers of the R and Phi sensor on a module differ by 64, thereby implicitly defining a module by this difference. Our aim should be get rid of all implicit assumptions like this. This will require extending the Detector Element interfaces.

Geometry-aware Coordinates

What is this?

Usually a geometry region in the pattern recognition is useful only if there are measurements (coordinates) associated with it. Typical requests are "give me all clusters on the phi sensor on this module compatible with the sector of a this r cluster". The purpose of geometry-aware coordinates is to answer these requests efficiently.

How and where to implement them

This is a bit less obvious than for geometry regions. To be efficient there has to some kind of caching strategy. Which is the best depends on the use case. This will become more clear once we identified the time critical use cases. As for where, I would say they should not be internal to an specific algorithm package.

-- Main.krinnert - 02 Apr 2007

This topic: LHCb > ATeamGeometryRegions

Topic revision: r1 - 2007-04-02 - KurtRinnert



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback