BornAgain - Refactoring #233

Simplify Geometry classes

21 Mar 2013 11:22 - wuttke

Status:	Resolved	**Start date:**	21 Mar 2013
Priority:	Normal	**Due date:**	
Assignee:	wuttke	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Seems, of the six include files (BasicVector3D.h Plane3D.h Transform3D.h Normal3D.h Point3D.h Vector3D.h) only two (BasicVector3D.h Transform3D.h) are actually used. Also seems, BasicVector3D is used in the sense of Vector3D; to forbid this abuse, BasicVector3D should be virtual.

I question the basic idea of CLHEP/Geometry to encode semantics (point vs vector vs normal) by defining almost identical classes Point3D Vector3D Normal3D that only behave differently when subjected to a transform. This leads to complications like the following:
Point3D p,q;
p-q; // is a vector, not a point
(p+q)/2; // center of gravity, hence a point?

I would rather have *one* vector class and different transform methods according to the situational meaning of the object.

History

#1 - 22 Mar 2013 11:45 - wuttke

- Assignee set to wuttke

We will not use CLHEP/Geometry in its present form as our generic library for 3D geometry. Therefore, I threw away the classes Vector3D, Point3D, Normal3D, Plane3D that are not used in BornAgain.

Next step: let's rename BasicVector3D into simpler Vector3D. Or Vector3 ? Or Threevector ? Other ideas ?

#2 - 23 Mar 2013 13:04 - wuttke

Still to do: eliminate translational component from Transform3D.

#3 - 30 Mar 2013 13:11 - wuttke

- Status changed from New to Backlog

#4 - 30 Mar 2013 13:12 - wuttke

- Status changed from Backlog to Sprint

#5 - 30 Mar 2013 13:17 - wuttke

- Status changed from Sprint to Resolved

Transforms are now methods of Transform3D instead of BasicVector3D.
This allows for meaningful inheritance Transform3D -> Rotation3D -> RotationSimple3D -> RotationZ_3D,
and for simplified computations in the more specific classes.
However, so far I did not implement the two intermediate classes.
This resolves the issue, except for the renaming of BasicVector3D, which goes to
the renaming campaign #240.

The most important aspect of this refactoring is probably the systematic use
of shared pointers (PTransform3D) which IMO simplifies the code enormously.