BornAgain - Refactoring #162

Refactoring: investigate removal of ISingleton classes

28 Jan 2013 10:40 - pospelov

Status:	Archived	**Start date:**	28 Jan 2013
Priority:	Normal	**Due date:**	
Assignee:	herck	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Sprint 13		

Description

Discuss if we really need singletons in our library and if not, remove them.

History

#1 - 11 Apr 2013 16:29 - herck

- Subject changed from Refactoring: get rid from MaterialManager singleton to Refactoring: investigate removal of ISingleton classes

- Description updated

- Status changed from New to Sprint

- Assignee set to herck

- Target version set to Sprint 13

#2 - 16 Apr 2013 13:33 - herck

After analysis of the evilness of singletons, its alternatives and the specific places where we use them, I suggest the following refactorings:

- FunctionalTestFactory and SampleFactory are ordinary factories (not Abstract Factories); this means we can easily replace the current Singleton implementation with just a static class implementation (or namespace with global factory functions).
- MaterialManager: here it seems that an instance of the manager should only be unique with respect to a single 'working environment' (which is not yet present in the current framework, but these would resemble the workspaces of LAMB for example); enforcing the uniqueness with Singleton is thus not really a good idea; my suggestion is to put a MaterialManager object inside a SystemComponent object; such a SystemComponent object would encapsulate the system state for an application instance (or workspace) (note: the use of such a SystemComponent enables the removal of any state in the application's components)

A final remark: in none of the above situations was there any polymorphism in play; so certainly no need for singletons then...

#3 - 16 Apr 2013 17:38 - herck

Actually, FunctionalTestFactory is better seen as a registry, which is initialized in main() of App and only passed to AppOptionsDescription. In this sense, users could create different sets of functional tests they want to run.
The use of SampleFactory as a globally accessible registry for some standard samples is left untouched for now. It only seems to be used for some non-IsGISAXS functional tests.
MaterialManager refactoring is also delayed to a new backlog item.

#4 - 16 Apr 2013 17:49 - herck

- Status changed from Sprint to Resolved

#5 - 03 Jun 2013 10:51 - herck

- Status changed from Resolved to Archived