Subject: Re: Proxy dict successfully used for the event loop
Posted by Bertram Kopf on Thu, 05 Nov 2009 13:34:39 GMT
View Forum Message <> Reply to Message

Hi Mohammad,
Mohammad Al-Turany wrote
Now to the general questions, this stuff is a copy of the BaBar stuff, do we have anybody who really can maintain this, modify it and develop it further?

This stuff is well tested and seems to work also in the new software environment. Since it is (or will be) part of the internal PndRoot software, it is possible for us -if needed - to modify and develop it further. The maintenance is also not difficult, since the code doesn't depend on anything (apart of the std template library) and is just written in c++ utilizing a couple of templates.
BTW: Why do we use Geant3? It is not maintaining anymore. And it is even an external package which we can not easily modify or maintain.

Mohammad Al-Turany wrote
Looking to the example task, it is a FairTask but without Init, and just create a dictionary in the exec (i.e. for each event) is that what we need, an event by event dictionary?

Yes! This is exactly that what I wanted to introduce. An event lifetime dictionary and in addition a job lifetime dictionary (not yet implemented).
The purpose of the event lifetime dictionary is to store and get access to objects which only exists within one event. The proxy takes care of deleting all pointers which are put into this dictionary.
The major purpose of the job lifetime proxy dict is to store and get access to condition objects which do not necessarily change from event to event. In principle, this proxy dict should be used as an interface to the conditions database.
I think, another advantage is, that this proxy provides the data in a type-safe way. Therefore error-prone casts are not needed in our software code.

Best regards,
Bertram.