

---

Subject: FairMCPoint

Posted by [Volker Frieze](#) on Tue, 06 Mar 2012 11:14:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I would like to trigger a discussion of the data classes implemented on the fairroot level, and start off by the class FairMCPoint.

This describes the geometrical intersection of a MCTrack with some sensitive volume. Quite obviously, it must contain three coordinates (floats) and a reference to the MCTrack (integer). In addition, one might want to store the momentum components of the track at this point - three more floats. It makes also sense to have some energy loss and track length. With that, one ends up with 8 floats and one integer - 68 bytes in total.

The size of the current FairMCPoint, however, is 224 bytes, which means that the framework introduces an overhead of some 230%. This is mainly due to the inheritance caused by FairLink.

Objects of the class FairMCPoints are very frequent, and the use of FairMCPoint is obligatory.

The facts above bring me to the opinion that the current implementation is not well architected. I would thus like to bring forward some theses for the discussion:

The data members of the base class FairMCPoint are legacy from the times when fairroot/cbmroot was developed for a single experiment.

FairMCPoint should not have any data member. Functionality needed for the framework (e.g., event display) should be implemented by abstract accessors.

The application (experiment) should decide which data members to implement in the concrete class, and in which representation (e.g. double, double32, integer).

This, by the way, holds for all data base classes (e.g., MCEventHeader, MCTrack, Hit, Digi etc.).

Making FairMCPoint derive from FairTimeStamp is counterintuitive and not a good design. A 3-d space point is not a special type of a time stamp.

Time stamps on the MC level do not make sense.

While FairLinks is a very useful scheme, the decision whether to use it for a particular data class should be left to the application, which can include the functionality either by making FairLink a member of its data classes, or by multiple inheritance.