Contribution ID: **197** Type: **poster**

Specifying Selection Criteria using C++ Expression Templates

Thursday 30 September 2004 10:00 (1 minute)

Generic programming as exemplified by the C++ standard library makes use of functions or function objects (objects that accept function syntax) to specialize generic algorithms for particular uses. Such separation improves code reuse without sacrificing efficiency. We employed this same technique in our combinatoric engine: DChain. In DChain, physicists combine lists of child particles to form a list of parent hypotheses. E.g., d0 = pi.plus() * K.minus(). The selection criteria for the hypothesis is defined in a function or function object that is passed to the list's constructor.

However, C++ requires that functions and class declarations be defined outside the scope of a function.
Therefore physicists are forced to separate the code that defines the combinatorics from the code that sets the selection criteria. We will discuss a technique using C++ expression templates to allow users to define function objects using a mathematical expression directly in their main function, e.g.,
func = (sqrt(beamEnergy*beamEnergy* - *vPMag*vPMag) >= 5.1*k_GeV).

Use of such techniques can greatly decrease the coding 'excess' needed to perform an analysis.

Author: JONES, C. (CORNELL UNIVERSITY)

Presenter: JONES, C. (CORNELL UNIVERSITY)

Session Classification: Poster Session 3

Track Classification: Track 3 - Core Software