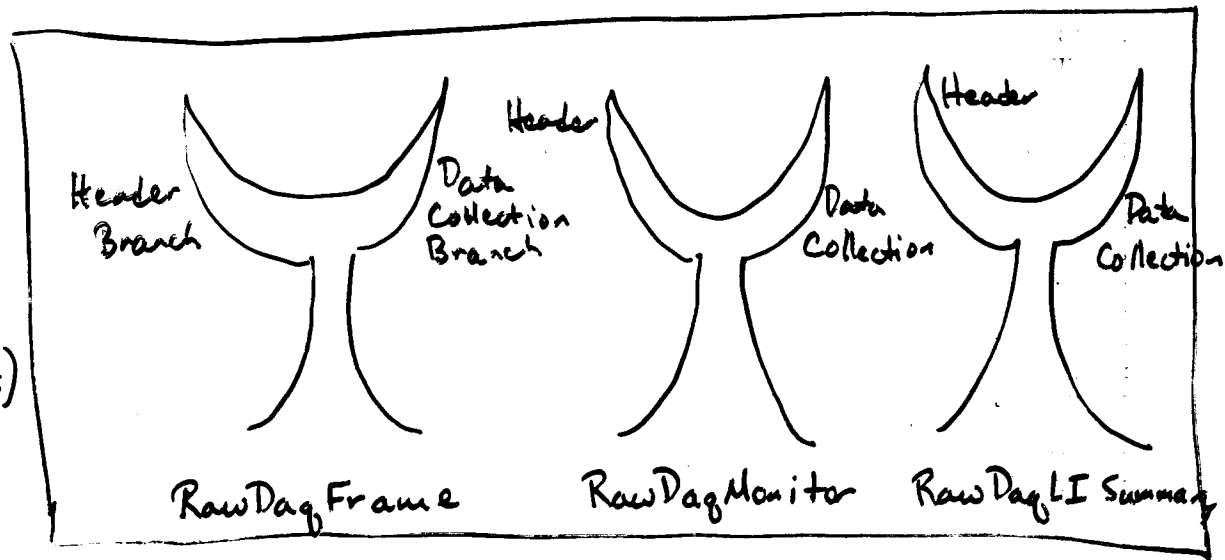


Root File Creation

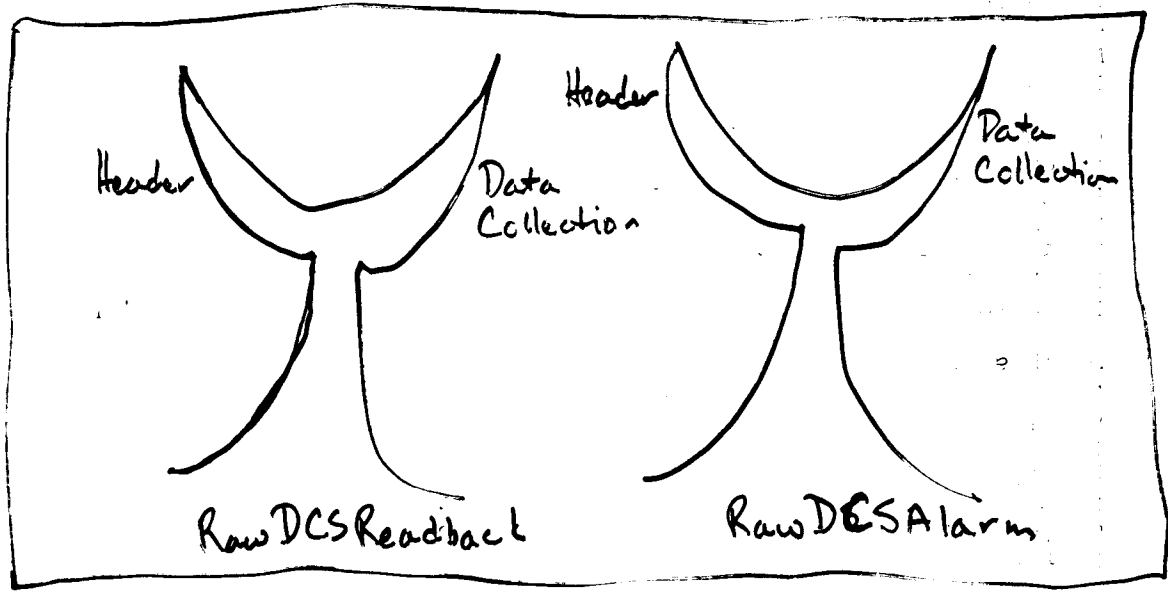
- The problem of writing the raw data out to file is a 2-stage process:
 - 1) The RawRecord format needs to be defined for each of the data types and these objects need to be filled at Run time.
(Robert)
 - 2) The filled RawRecord objects need to be ~~added~~ stored in their respective ROOT TTree data structures and the TTree's need to be written to file. (Sue)

Robert's proposal for the number and structure of ROOT Tree's in the 2 raw data files:

DAQ File
(3 TTree's)

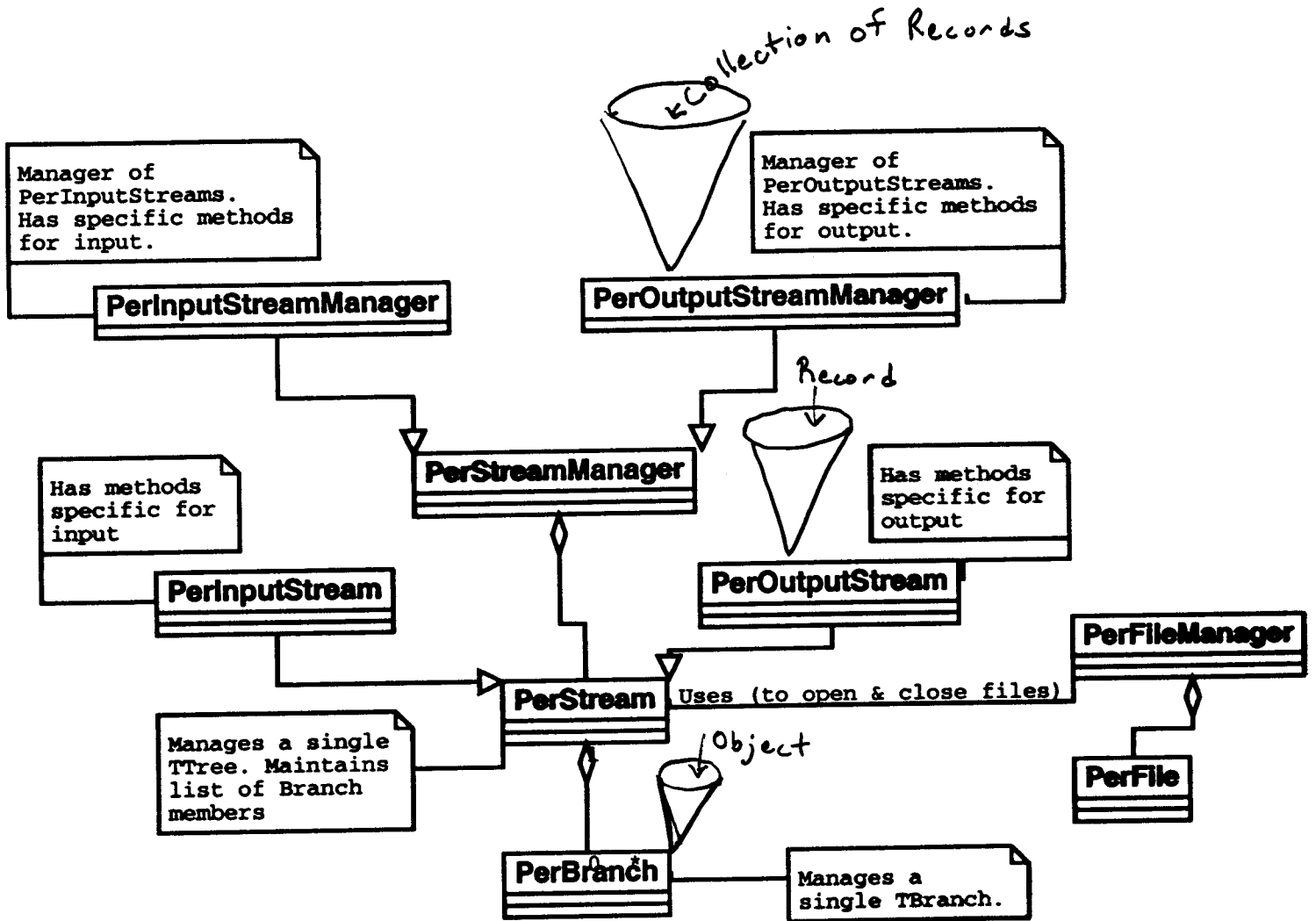


DCS File
(2 TTree's)



- This problem of storing the raw data to file is a specific use case of a Persistency package that I have been writing for the offline system.
- The purpose of the Persistency package is to manage the input/output of data to permanent storage (i.e. a ROOT file on disk).
- The persistency package provides tools to:
 - 1) Manage the I/O of data to multiple (1:n) TTree's each with 1:n Branches.
 - 2) Manage the 1:n files in which the ROOT TTree's are stored.
(The managed TTree's can share files with other TTree's or be stored individually in separate files.)

Persistency Stream and File Management Classes



↳ Push to ROOT TTree branch on "store"

A Stream ≡ 1 ROOT TTree.

Status

- A first working version of the output stream management and file management classes have been submitted to the offline software repository and are available for use.
- I've written a DemoDagOutputModule class and DemoDag driver to illustrate how the persistency classes can be used to write the raw data to file (also submitted to the repository).
- Mark Messier has integrated these ^{Persistency} classes into the offline job framework so that we can write MC raw data to file, (RawDagFrame records only).