Thoughts on pipeline post Data Handling Meeting

Here are what I take to be the primary issues that need addressing in the next iteration of the pipeline:

High level

- more easily debuggable code; a language with a debugger. We've decided the bulk of the code that performs logic will be in java. Perl (or python) would be used for those functions closely tied to the unix O/S.
- should consider providing a facility for general users (hence user roles needed to control access)
    - segregate resources - disk, batch allocation, db table space?
- should consider running the system tests via the pipeline - also segregate its resources?)
- be able to run on remote (to SLAC) farms, maintaining the bookkeeping of the output
    - need interfaces to main resources - database and batch system. Java's hibernate provides the database interface.

Mid Level

- the graph of processes that can be supported needs to be much richer than provided now
    - conditions for running a process need not be restricted to availability of datasets
    - multiple processes should be able to work in parallel on a given dataset
    - a process may depend on multiple datasets
    - will presumably need a mechanism to split large datasets for faster parallel processing, then reassemble the products
    - a task can take inputs from another task
    - the graph should support versions of process runs and if desired follow up subsequent processes whose inputs have now been incremented. This could allow reprocessing while maintaining the basic identity of the original task.
- the bookkeeping should keep track of the important applications being run (eg in current MC, version of script is not so important; version of GlastRelease is, but is not recorded)
- allow code patches, but record that they are in use. Presumably on a process instance basis (rather than by some time stamp after which it is assumed all code uses this version?).
- all files produced by the pipeline should be archived, and for real data, the input files too. At least the tasks run for the public good. Maybe not for users.
    - at present, the archive is complicated by parallel tasks operating on a single run for I&T. Not clear how this would evolve in the world of a richer graph structure.

Details

- outtput files should be write-protected

More to come