Cornell University
Laboratory for Elementary-Particle Physics

CHEP 07
Christopher D Jones, Andrew Dolgert

HEPTrails An Analysis Workflow and Provenance Tracking Application

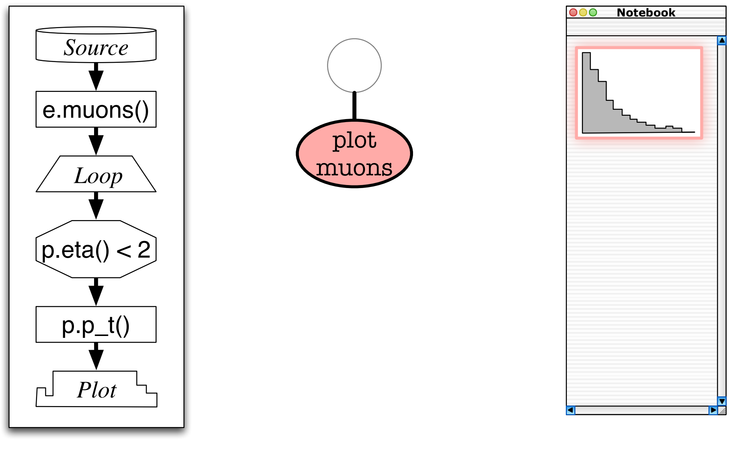
Know What You Did

For each change of a workflow, a new version is added to the version graph. The results of the workflow are displayed in the notebook. Each entry in the notebook can be associated back to its original workflow.

E.g., Plot muons

For each event, get the list of muons, loop over that list and for each muon, if it passes the eta cut, plot the muons p_t

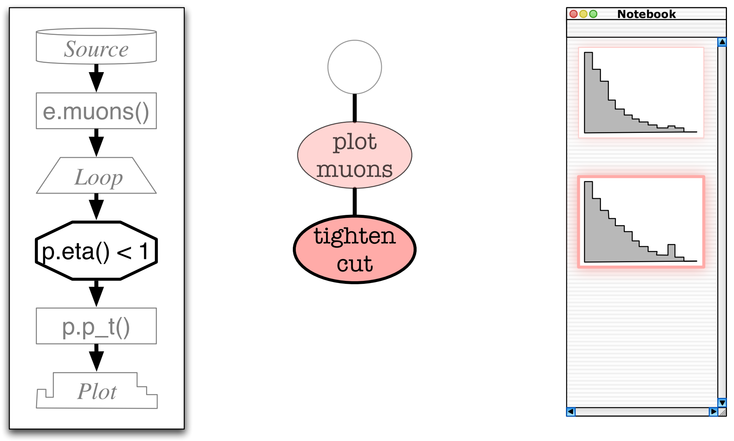
Workflow **Versions** **Notebook**



E.g., Tighten cut

Tightening the cut on eta will cause a new version to be added to the version graph with the associated plot added to the notebook.

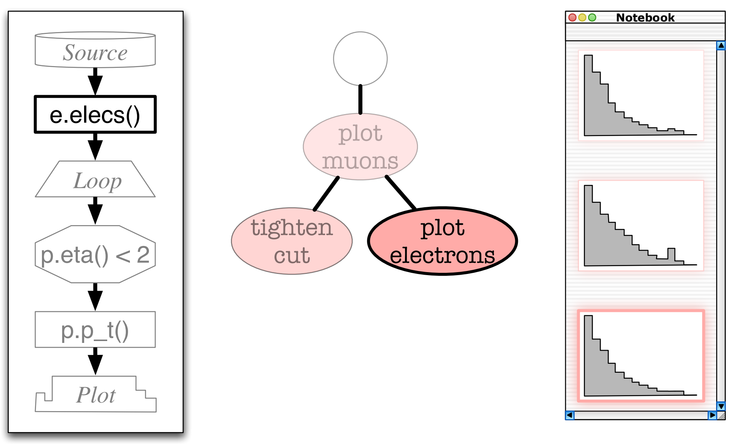
Workflow **Versions** **Notebook**



E.g., Plot electrons

Going back to the original muon workflow and changing it to plot electrons instead causes the new version to be associated with the original version.

Workflow **Versions** **Notebook**



Streaming Workflow

HEP analyses typically want to apply the same workflow to each Event. Even within an Event one typically applies a sub-workflow to elements of a list. A streaming workflow is then a workflow which can be applied to elements of an incoming 'stream'.

Simple building blocks

A workflow can be assembled by connecting together simple modules whose attributes have been tailored for the particular workflow. The modules do work at a very fine-grained level, equivalent to a line of C++ code.

Source — Provides the Events to process
E.g., provide the indidividual entries from a ROOT TTree

Transform — Converts the input item to something else using a rule
E.g., select the muon list from an event
E.g., calculate the invariant mass of a particle

Filter — Only passes input item to later modules if item passes the criteria
E.g., select only items with a small eta

Loop — Items in input container are streamed out individually to modules later in the workflow
E.g., stream each muon from the muon collection

Accumulate — Output is not generated until all input items in the sub-stream have been examined
E.g., find the maximum momentum muon
E.g., accumulate all input items into a new list

Plot — A specialized Accumulate which plots its inputs

Binary — Takes two inputs and applies a transformation. The rate of output depends on the rate of the input streams.
E.g., take two list and do combinatorics

Optimize by generating Python

Using modules when actually running the workflow was found to be too slow. Instead, the modules each contribute Python code to form a python program which is equivalent to the requested workflow. This python code is then executed.

E.g., auto-generated Python for muon plot

```
source      = Source(...)
transform1  = lambda e: e.muons()
filter1     = lambda p: p.eta() < 2
transform2  = lambda p: p.p_t()
plot1       = Plot(...)

plot1.reset()
for e in iter(source):
    trans1 = transform1(e)
    for p in iter(trans1):
        if filter1(p):
            trans2 = transform2(p)
            plot1.accumulate(trans2)
plot1.done()
```

Don't Make Me Wait

GUI on separate thread

Edit a new workflow or add notes to the notebook while a workflow is executing.

Preliminary results

While a workflow is running, you see the histograms updating. If you see a problem, you can stop the workflow.

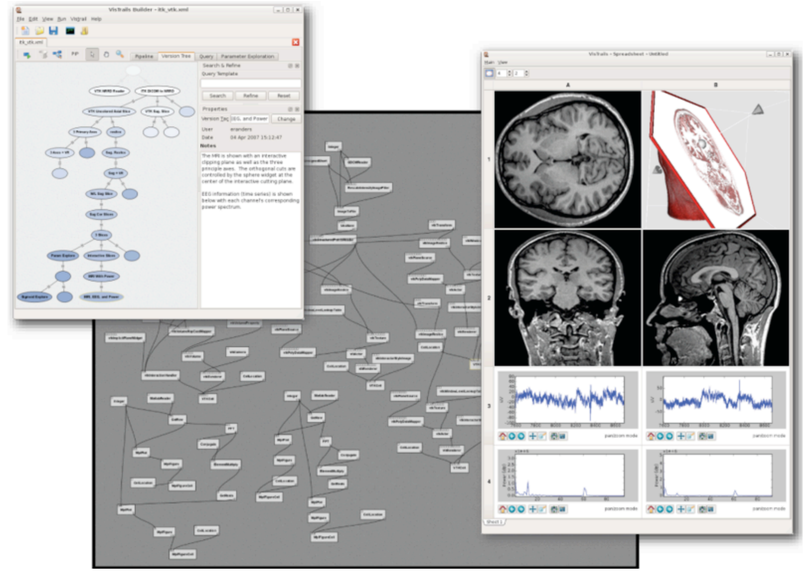
Multiple simultaneous workflows

If you have a new idea while one workflow is running, you can create and start additional workflows without waiting for the first one to complete.

Based on VisTrails

http://www.vistrails.org

VisTrails is a scientific workflow and provenance management system developed by the University of Utah's Scientific Computing and Imaging Institute which supports data exploration and visualization. The program is written in Python using the Qt GUI toolkit.



The design of VisTrails separates the workflow construction GUI from the visualization GUI (which uses a spreadsheet metaphor), the provenance tracking system and the workflow execution engine. This allows us to keep the workflow construction GUI and provenance tracking system but replace the available modules and the workflow execution engine as well as provide a new notebook-style visualization GUI capable of showing preliminary results.

A Work In Progress

The present version of the program is a 'working design sketch' which is being used to explore the technical and user interface challenges of such a system. Future work includes:

LAN workflow farm auto-detect and distribute workflow to a local area cluster

automatic caching save intermediate results from a workflow for use in other workflows

declarative workflow language construct a workflow textually

NSF