Analysis Tools

A brief overview of currently available tools

How to choose?

ATLAS Software Workshop
20.05.99

M. Stavrianakou

The market yesterday and today...

- CERNLIB components
- LHC++ components
- ROOT
- JAS
- OpenScientist
- Other (sometimes with limited functionality):
  HippoDraw, Grace, various commercial tools...

How to choose?

- Evaluate available/appealing tools according to criteria based on:
  - Requirements and use cases
  - Architectural choices: components vs framework
  - Choice of standards
  - Choice of programming languages and technologies
  - Resources: free, commercial, in−house...
  - Timescale: today, next year, in 2005?

Functionality to be considered in evaluation (I)

- I/O and data models
- Histogramming
- "Tupling"
- Minimisation and fitting
- Plotting
- Interactivity
- Other

Functionality to be considered in evaluation (II)

- I/O (Objy, light−weight persistency, ROOT, RIO,...)
  - choices depend on data types and volumes, as well as access and selection patterns
  - should be largely "decoupled" from "real" analysis functionality
  - ideally it should be possible to use different schemes depending on the task e.g. access raw data from an Objy database, store "ntuples" or histograms in "files" using light−weight persistency schemes, carry out "stand−alone" analysis, store "final" objects back in "main" Objy database

Functionality to be considered in evaluation (III)

- Histogramming (HTL, ROOT histos, JAS, OpenScientist...)
  - Some will soon be interchangeable
  - Some offer "new" functionality (e.g. user defined partitions, some level of intechangeability between persistent and transient histograms and histogram factories in HTL)
  - Histogram link to "raw" data it was derived from may be important: feasibility must be demonstrated with realistic analysis scenarios

Functionality to be considered in evaluation (IV)

- Tupling (HEPOBMS event collections and tags, ROOT Trees and Ntuples, HepTuple...)
  - Can ntuples (PAW, ROOT) be replaced by HEPODBMS events collections and tags? First studies seem promising (see presentation by S. Resconi on A−>ττ−>jet−lepton−pTmiss analysis http://wwwinfo.cern.ch/asd/lhc++/meetings/980701/atlas/index.htm) Feasibility must be further demonstrated with realistic analysis scenarios and detailed (meaningful) comparisons must be made...

Functionality to be considered in evaluation (V)

- Minimisation and fitting ((wrapped) Minuit, GEMINI (Minuit or NAG and HEPFitting), commercial or custom fitters...)
  - Must (and could?) be largely interchangeable
- Plotting (PAW, ROOT, HEPInventor/HEPExplorer, JAS, OpenScientist...)
  - 2D extremely important and rather poorely addressed by commercial tools
  - PAW/ROOT paradigm very appealing
  - Interchangeability, in any case, very desirable

Functionality to be considered in evaluation (VI)

- Interactivity (PAW, ROOT, HEPExplorer, JAS, OpenScientist...)
  - PAW/ROOT paradigm very appealing
  - HEPExplorer so far failing to convince the end user
  - JAS being considered as an alternative
  - Open Scientist promises flexibility
- Powerful GUIs may not be able to replace full scripting functionality
  - but how to choose a scripting language?
    - SWIG to generate Tk/Tcl, Perl, Python increasingly used
    - CINT appealing to some, controversial to others...

Other issues to be considered in evaluation

- Useability and Performance
- Modularity and Flexibility, Maintainability and Extensibility
- Replaceability
- Restrictions imposed by the choice of languages, standards and technologies e.g. Raw data in C++ vs analysis in Java, use of Corba
- Restrictions imposed by the legacy software (F77, C, other)
- Resources and timescales

Some comments

- ATLAS should decide which tools to evaluate depending on the requirements, functionality, performance, software quality, overall architecture and standards, resources and timescales (not necessarily in that order!)
- BUT a serious LHC−, or CERN−, or HEP−wide effort should be made to coordinate (wherever feasible) analysis tools work from the providers side (LHC++, ROOT, JAS, OpenScientist, other) to avoid duplication and waste of effort and to maximise interoperability; this should be largely based on user requirements and feedback.

Plans

- Start Requirements Review
- Start evaluation exercises
  - Identify realistic use cases e.g. based on
    - 1TB database
    - Combined reconstruction ntuple
    - ATLFAST++
    - test beams
    - simulation
    - ...
  - Establish priorities, identify manpower
  - Start the real work!