PASFRG Conclusions on some technical issues

- Data Formats
- Scripting Language vs Programming Language
- Scripting language identical to some high level programming language??

Data Formats

- Input: Import of arbitary foreign formats
  - The analysis tool must supply user hooks for user written conversion routines to read foreign data formats
- Output: Creation of private data formats
  - The analysis tool must support data formats that are optimized for efficient access based on user access patterns, including event selection, “bank” selection and data arrangement

Scripting vs Programming Language

- Fundamental requirement is an interactive tool to access data and develop physics algorithms
- The scripting language must support:
  - Data access
  - commands
  - simple evaluation functions
  - invoking high level language procedures (pre-compiled or dynamically linked)

Scripting vs Programming Language (cont)

- Scripting language should support:
  - flow control, looping and complex evaluations
- “COMIS like functionality is unnecessary once the scripting language supprts dynamic linking”
- A separate programming language is not required

Suggested paradigm for interactive algorithm development

- Run scripting language in analysis tool
- In a separate window edit your high level language routines
- Dynamically link in HLL routines
- KEY REQUIREMENT: scripting language must support the offline object model for data

Should scripting language be "identical" to some high level language?

- For portability?
  - not an issue once dynamic linking is supported
- For ease of use (not having to learn a new language)?
  - not an issue: need to learn command syntax anyway

Script =? HLL (cont)

- Scripting language will never be identical to HLL
  - different bugs, unknown subset of HLL very dangerous
- C++ syntax (invoking methods of some object) not viewed as a natural way to express commands (or to learn C++)
- No requirement for scripting language to be a HLL