ROOT Viability
Post-(Pre-Evaluation)
Take 2

PasRec Meeting
28 July 1998
Rob Kennedy

- **Short-term Support Issues**
- **Other Support Issues**
- **Short-term Functionality Issues**
- **Conclusion**

This is a brief list of known issues brought up by comparing experience with ROOT to the evaluation criteria from PasFrg and PasSuma. Some issues I recommend be considered sufficiently important ("short-term") that we should consider making a recommendation contingent on their resolution in the short-term.

ROOT - Short-term Support Issues

➪ Background: I am building ROOT v2.00.08 for use with Kai C++ under Linux RH 5.0 (binaries are not even distributed) and under IRIX6 (only stripped, optimized binaries are available). Goal is to use ROOT Event I/O to save/restore CDF event data. I am attempting to link to ROOT libraries in SoftRelTools environment.

➪ Issue: Code Maintenance - ROOT is still only supported under CMZ (child of Patchy and Zebra). Local maintenance and builds are *very* tedious and difficult, requiring knowledge of CMZ to do it correctly. Some makefiles are becoming available for v2.00.09, but only for a few system-compiler combinations (none for Kai). For example, the Linux2-KCC v3.3 makefile and build script is untested since their Linux Kai evaluation license expired.

➪ Reaction 1: Get ROOT team an account on a CDF Linux platform (cdfpca?) with Kai v3.3 installed, as was done on cdfsga Kai v3.2

➪ Reaction 2a: Require a public CVS-based repository (like egcs)

➪ Reaction 2b: Require Makefile-based build procedures

ROOT - Short-term Support Issues*

➪ Issue: No official commitment to support ROOT
➪ Reaction: FNAL management needs to work on this to insure ROOT does not "creep into" FROOT through lack of external ROOT support. We should at least require a ROOT team presence here in the short-term to deal with the highest priority issues.

➪ Issue: No official commitment to support CINT. How to maintain, improve, and ***document*** it? Cint is not just an interactive scripting language. CINT is also used to build object dictionaries to support ROOT's form of RTTI, such as how D0 applies CINT within D0om.
➪ Reaction: Who will commit to improve and extend ROOT's CINT? FNAL needs to consider one of the following options:
(1) replace CINT outright (2 FTE years?, EDG C++ front-end), in collaboration with the ROOT team, if agreeable to ROOT users,
(2) negotiate a support agreement with ROOT team specifying precisely the "C++-like" language to be supported for FNAL use,
(3) use a command interpreter with ROOT other than CINT, such as Python, and drop interactive ROOT RTTI (batch CINT kept)???

ROOT - Other Support Issues

➪ Issue: ROOT does not now support C++ exception-handling, a feature required by D0om. This makes ROOT supplied libraries link-incompatible with D0 SoftRelTools-built libraries.

➪ Reaction: Does D0 want to link to ROOT libs on September 1?

➪ Issue: ROOT includes significant functionality which cannot easily be replaced by existing implementations (Linear Algebra), or is implemented in (== supported in) Fortran (converted to C).

➪ Reaction: ? We should address this for the long-term. ROOT carries MINUIT along with it... so pieces of "Cernlib" live on.

➪ Misc. Issues: ROOT crashes under Linux2-KCC v3.2. ROOT memory management does not work on some system-compiler combinations, preventing use of shared memory with ROOT.

➪ Reaction: ROOT works under Linux2-KCC v3.3, to which CDF/D0 will be moving in the near future. CDF/D0 do not seem to be using the platforms on which ROOT memory mgmt fails in some way. Remember this, though, when new platforms are considered.

ROOT - Functionality Issues

➪ Background: Tom Phillips and new grad student (Duke/CDF) have volunteered to help evaluate ROOT from a Paw user's, new user's point-of-view. Gordon Watts applied ROOT in study (next).

➪ Issue: New users: No window auto-refresh. One must select [Quit ROOT] twice to exit ROOT's "hello world" test program. CINT has a non-C++ command language (should be wrappered in a GUI). ....
➪ Reaction: We should follow up on these in the short-term if for no other reason than to insure we recommend something which is friendly to new users. Paw should *not* be our benchmark for that.

➪ Issue: ROOT supplies an HBOOK to ROOT Ntuple converter program (w/ compression --> smaller output), but there is no tool to help re-use kumacs in the short-term. Kumacs record analyses too.
➪ Reaction: We or ROOT can create a means to compile and use COMIS functions, fill common blocks from Ntuples, and allow some re-use of kumacs/Comis. We should have a plan which keeps users productive in the short-term while they learn C++ and ROOT.

ROOT - Functionality*, Viability

➪ Issue: Gordon Watts: Difficulties w/reasonable, non-trivial tasks
http://www-d0.fnal.gov/~gwatts/talks/root-thoughts-020298.ps

➪ Reaction: Many issues he raises may be added to our high priority list, and nearly all made part of the evaluation. IMHO, His background is closer to a majority of *near future* FNAL Root users.

➪ My considered opinion: ROOT is viable as a candidate for a short-term, as well as a long-term, physics analysis software toolset. The issues surrounding CINT are especially worrisome, since CINT is crucial to much ROOT functionality, will be expensive and risky to replace. We should make any recommendation of ROOT (if selected) contingent on all high priority issues being resolved, or a definite plan of resolution being developed, in the short-term.

➪ We need a broad set of evaluators since some issues were only noticed when someone did something *other than* run the ROOT demos (as I did for initial evaluation). Folding in experience of Scott Snyder, Gordon Watts/Brown, and Duke/CDF folks will help.