CDF/MEMO/SCMB/PUBLIC/4780
October 16, 1998
Version 0.5

Preliminary Evaluation of ROOT for CDF from a Users Perspective

Michael Kirby
Thomas J. Phillips

Abstract

We have installed ROOT at Duke and used it for some simple tasks in an effort to evaluate the package from a users perspective. We find a substantial gap in the documentation between the tutorials and the very detailed object descriptions. We make a number of suggestions for additions to the standard distribution package, including some suggestions for packages that would aid in the transition from PAW to ROOT.

1 Introduction

In an effort to keep up with the developments in analysis software and prepare for Run II at Fermilab, the High Energy Group at Duke University has begun using the ROOT [1] software/framework. The main purpose was to try to evaluate the system from an end user perspective and at the same time learn the capabilities and syntax.

Most of this work was done by Michael Kirby, who is a beginning graduate student with no PAW experience. Michael also has some experience with object-oriented programming and limited experience with Fortran. In addition, another graduate student from the Triangle Universities Nuclear Laboratory at Duke has also been working with and learning ROOT and helped in the evaluation process and with general problems.

The majority of the work was done with an HP 9000 workstation running HP-UX 10.20. ROOT appears to be extremely flexible and powerful while slightly elusive for a first time user. No serious deficiencies were found within the software, but there were several places where a change of design could be quickly implemented would greatly improve the ease of use. The documentation contains a great number of examples, yet lacks thorough explanations. ROOT appears extremely well designed but still needs some refinement in order to make it as smooth as possible.

2 Installation

ROOT is available from the web site at `http://root.cern.ch/` and comes completely contained in the appropriate tar files. Installing the software is as uncomplicated and as easy as possible. ROOT is packaged extremely well, and comes ready to run "out of the box," without the need for any compiling. The web page at CERN makes the process of getting all the information and the correct binaries incredibly fast and easy. The system was installed on three different platforms (HP-UX, Solaris, and Linux) and ran without trouble or the need for any additional configuration.

One difficulty was discovered with an associated program that is designed to help in the transition from PAW (HBOOK) to ROOT. The utility "h2root" did not run due to a library file that it could not find. The installation files say that for version 2.00/2.08 there is no need to have a declaration pointing to the shared libraries. While this is true for the ROOT program itself, which runs fine without any such declaration, it is necessary to define the variable SHLIB_PATH in order for the h2root utility to work. The task of defining the variable is not difficult, but either h2root should be fixed or the comment in the installation file removed or edited. This was the only problem with the binaries, as everything else ran just as advertised.

3 The Learning Curve

The first time a user opens ROOT, they are face to face with CINT, which at times is a serious challenge. This command line C interpreter is an excellent shell and contains several very well designed ideas. However, the shell comprehends only pure C and C++ code which can be both a great advantage and a hindrance.

When first using ROOT, one of the first things you must learn is that all general commands must be preceded by a leading ".". Due to the design of CINT, this is necessary (assuming that the leading period means that the statement that follows is a call to a function in the "default" object), but it should be noted that placing the leading "dot" was not obvious to either of the evaluators. It was never explicitly explained why CINT works this way.

The one place that mentions the leading dot is the CINT help page [2], which is in need of some revision and expansion. The most obvious question is that, with the very thorough web based documentation of the ROOT classes, why the CINT help file was not more complete. The possibility of several examples does not seem to be beyond reasonable for some of the more complex commands, along with an explanation of their output.

The final comment about CINT is that as a pure C/C++ interpreter, the program does not make for an easy transition for Fortran/PAW users. The advantages of having the interface using the syntax of a high level language are vast, yet, at the same time this makes the program so cryptic and foreign to a non-C programmer that the learning curve is very long. I do not believe that the structure should be changed but at the same time I think that there should be a strong emphasis on the fact that knowledge of C++ is a must for efficient use of ROOT. The claim in the mission statement that ROOT offers an ideal environment for learning C++ seems a bit strong, but it does seem to be excellent as a first opportunity for serious programming after learning the language. Anyone hoping to use ROOT needs to learn C++ first in order to avoid great frustration and time consumption.

4 Weaknesses and Suggested Improvements

In the process of using ROOT we found a number weaknesses that suggest improvements to the program that would make it more user friendly:

- CINT lacks a functioning pager. Whenever the help text is called up, the text scrolls a given number of lines (33) but does nothing to match the number of lines on the terminal. The use of a pager would help with not just the help file but also with the output of several CINT commands, such as the listing of the local and global variables.

- Although the CINT works extremely well after one is familiar with ROOT, there seems to be a need/demand for a GUI (graphical user interface) and several other added pieces of functionality. The TBroswer class provides a great way to process existing ROOT files except that in order to get the browser up and running one must enter C++ code. And simple as the one line code is, it was not obvious where one would find the class or how to implement it. This places the new user in the debilitating situation of facing a command line, when the demos show great promise of point and click interfaces and menus. The solution seems simple enough and that would be a short menu similar to the one used with the ROOT demos. If nothing else it would provide the first time user with an quick and easy way to get at ROOT files and some of the useful tools such as the graphics and histogram editors without the need of learning C++.

- An easier mouse event interface would be extremely useful. This difficulty became apparent while trying to write a macro that would integrate over a given range of bins in a histogram. (We note that this same topic has been discussed in the ROOT Talk Digest and will be integrated into the ROOT version 2.00/09.) Although this specific problem has been solved, the advantages of having the ability to easily use mouse events within macros make adding a mouse event class very desirable. This would be a great improvement for constructing advanced analysis programs.

Hopefully these request would be feasible and would certainly improve the user interface of ROOT.

There is a need for several prepackaged macros that cover some simple processes. Since ROOT is being considered as the replacement for PAW, it seems reasonable that ROOT have a package to make the transition easier and a package that would mimic some simple functions of PAW. Here are a few examples:

- While ROOT is distributed with h2root to translate hbook files into ROOT format, this only solves part of the problem of transferring an analysis from PAW to ROOT. Data in PAW Ntuples are often analyzed using COMIS fortran subroutines. It would greatly aid the conversion from PAW to ROOT if there were a standard package that could fill the PAWC common block and run a Comis routine for an Ntuple that has been converted into ROOT format.

- There is a standard PAW process for making an Ntuple from a standard ascii text file. This is a common problem, and users should be able to do this without needing to write a C++ routine.

- Lacking a small amount of prewritten macros, the idea of having to learn a new high level language and programming structure, will appear too daunting a task for many. Obviously, one cannot expect the process to be completely automated, but a set of macros that are fill in the blank seems simple enough and definitely worth while for the general end user.

These suggestions will help PAW users switch to ROOT quickly and easily continue their work without the detailed process of learning C++.

5 Documentation

The last and probably the most worrisome part of trying to work with ROOT is the documentation. It must be said that the developers deserve a great amount of credit for both the thoroughness and availability of documentation. Having the class structure listed on the web is extremely useful and easy to deal with. The difficulty though comes from the fact that this becomes almost exclusively the only documentation. There seems to be a serious gap between the idea of getting someone from a beginning user to a working programmer. The daunting task of reading the almost 700 pages of class definitions does not seem to be the right way to learn ROOT and it is quickly apparent that this volume should be used almost exclusively as a reference.

At the other extreme, the tutorials that are provided with ROOT are not as complete as possible. The serious lack of comments within the tutorials greatly handicaps someone trying to learn both how to use ROOT and to program C++. Of course, it is not the place of ROOT to include a C++ manual, but there should also be enough explanation in the code so that there is a strong delineation between what is a part of ROOT and what is part of C++. As an example the Object Oriented User Interface tutorial has three lines of comments for more than thirty lines of code. Reverse engineering the options for function calls is both difficult and inefficient. Hopefully with an increased amount of comments within the tutorials, they would become more than just a struggle with the code and help new users actually learn the possibilities of ROOT.

Another serious problem with the documentation is the lack of several items. One of the most frequently noted problems is that not all of the options for a given function call are listed or are so scattered as to be confusing. Specifically all of the possibilities of the standard Draw function never seem to be found. This may of course just be a consequence of my unfamiliarity, but at the same time the difficulty of tracking this information seems to make some modification necessary. The overall documentation for the framework is amazing both in volume and standardization, but this seems to be the area that needs the most work. In order to make ROOT work for the standard user, there needs to be something more robust than the current tutorials to act as a buffer between the beginning user and the magnitude of the class definition reference.

6 Summary

Using ROOT over the last several months has opened up the possibilities of having a smooth, powerful, and elegant analysis tool. The software itself seems to be designed amazing well and thoroughly so as to not just replace PAW but extend well beyond those capabilities. At the same time, we have found several places where the implementation of changes would not just improve the software, but improve the interface and ease of use for the physicist. Unfortunately the largest complaint focuses on the toughest of areas, documentation. Although extremely thorough and well thought out, there should most definitely be some serious revision and improve so that something as well designed as ROOT can be easily implemented. The framework has been great to work with and hopefully these suggestions will help to improve ROOT as it becomes part of the physics analysis community.

References

[1] (CERN, Rene Brun *et al.*)

[2] http://root.cern.ch/root/CintInterpreter.html