# History of Analysis Tools in HEP (a personal view)

Liz Sexton-Kennedy



HEP Software Foundation

hepsoftwarefoundation.org



### Not Grandparent, but a Dynasaur



Material comes from a talk written by Rene for CHEP 2012 https://indico.cern.ch/event/149557/contributions/1385791/attachments/150332/212934/chep2012.pdf



#### The Ancestry of ROOT



Rene drew associations between projects that were the forbearers of ROOT and GEANT



### The Ancestry of ROOT



I started in the field in 1983, packages like hydra, zbook, and zebra were developed to patch up missing language features: strongly typed structured data, and dynamic memory allocation at the cost of debug-ability (couldn't do array bounds checking).

As an infrastructure person I had fought too many memory over-write battles, FORTRAN90 was not real yet and it seemed the right time to press for change.



HEP needed Structures, References, Clear object ownership, and Reflection



I want to spend a slide on this...

ROOT caught on quickly in the field because it was a pinout replacement for PAW

It's user support was small but strong





#### There were 3 committees

- 1. A Technical Reports group
- 2. A maintainability group
- 3. A recommendations committee

There were 3 products seriously evaluated, ROOT + 2 others (MATLAB, FNAL in-house)

The biggest concern at the time was the maintenance of CINT

We recommend that ROOT be adopted as the standard physics analysis package for Run II, contingent on a collaborative agreement with the ROOT team. It should be recognized that this recommendation depends critically on timing and on sharing development with outside collaborators,...

It is highly likely that by the end of RUN II (or by the time of the LHC) that commercial components will be heavily used for analysis tasks. Commercial offerings should continue to be investigated and made available (perhaps on limited platforms). The Computing Division should also initiate formal collaboration with the LHC++ project so as to have some influence on the choices made and direction taken. These two initiatives, while lower priority than the immediate ROOT support and development needs, should position us to take full advantage of expected evolution of these products.



I think trying to integrate a not complete C++ interpreter makes no

sense and will not add any additional value to ROOT. We should work with  $% \left( {{\boldsymbol{x}_{i}}} \right)$ 

Masa in a parallel rewrite of the CINT kernel in C++ all the while keeping CINTs good features and backward compatability in mind. Having

spent yesterday a whole day in debugging CINT I agree that a rewrite would be nice (CINT clearly started as a much smaller project than it

currently is, i.e. from simple C subset interpreter to (almost) full ANSI C++ interpreter. It is already amazing that Masa managed to evoluate the code like he did). The ideal would be to have Masa + a VERY GOOD Fermi lab guy work for a year on this project. The benefits of a rewrite should be:

- robustness (better error handling for trivial typing errors)
- better template support
- more transparent, i.e. more maintainable code
- exception handling

all the while understanding that C++ is a huge monster of a language (largest and most complex computer language ever) normally only implemented by fairly large teams of people (as I know from HP experience). As always,implementing 90% of C++ (like we have now) will

be fairly well doable. Getting to 100% might be almost impossible.

- Jan 95: Thinking/writing/rewriting/more thinking
- Nov 95: Public seminar, presentation of ROOT 0.5
- Spring 96: decision to use CINT
- Jan 97: ROOT version 1.0
- Jan 98: ROOT version 2.0
- Mar 99: ROOT version 2.21/08 (1st ROOT workshop FNAL)
- Feb 00: ROOT version 2.23/12 (2nd ROOT workshop CERN)
- Sep 00: ROOT version 2.25/03
- Dec 00: ROOT version 3.00/01
- Jun 01: 3rd ROOT Users Workshop at FNAL
- ...
- 2005: ROOT finally becoming mainstream at CERN
- ...
- Mar 13: 8th ROOT Users Workshop Saas-Fee
  ROOT Users Workshop 11-14 March 2013

Fons

<sup>-</sup> thread safe



## <u>Questions</u>

Does this audience agree with Rene and I that "HEP needs Structures, References, Clear object ownership, and Reflection" ?

How can we make reconstruction and analysis work together better? Do we agree where the boundaries are?

When writing my presentation I had lots of push back about adopting the Apache model for new developments... is it really a silly idea?

Fons on ROOT: Does it not solve our current problems anymore, has it reached end-of-life, can it not anymore be upgraded to handle what needs to be handled next, has the support collapsed, does the community using it stop, are there already much better alternatives? As long as that is not the case ROOT should continue and, to the benefit of the user community, all should be done to make it even better, make it adopt the latest techniques so it can stay ahead of the technology curve (LLVM, ML, etc) while at the same time serve our community with a stable, solid and backward compatible, platform. The community however should make sure that ROOT's evolution continues, that latest techniques are adopted, that it is benchmarked against up and coming technologies so it will maintain its leadership role.