CINT Retrospect

ROOT Workshop Mar 2013

Agilent Technologies

Masaharu Goto

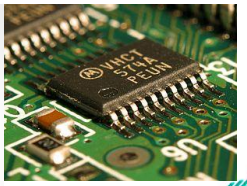
Software for Scientific Research

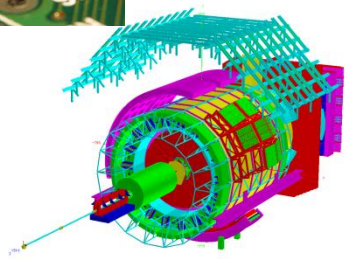
Fundamental Needs

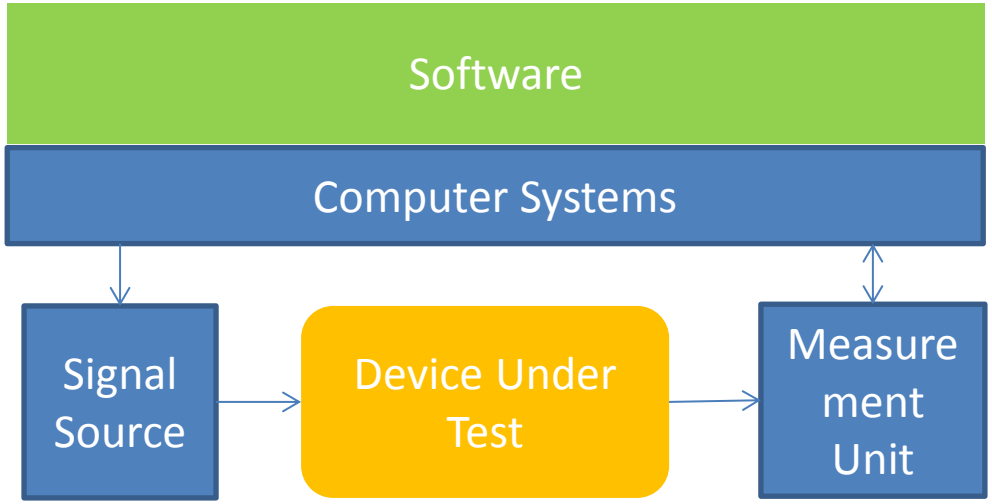
-Users are not necessarily software experts — Easy
-Flexibility to deal with unpredictable phenomenon — Interactive
-High performance computation for simulation — Fast
-Stable over long period of time — Sustainable
- Work with foreign library — Flexible

Interpreter for Standard Computer Language

DESIGN





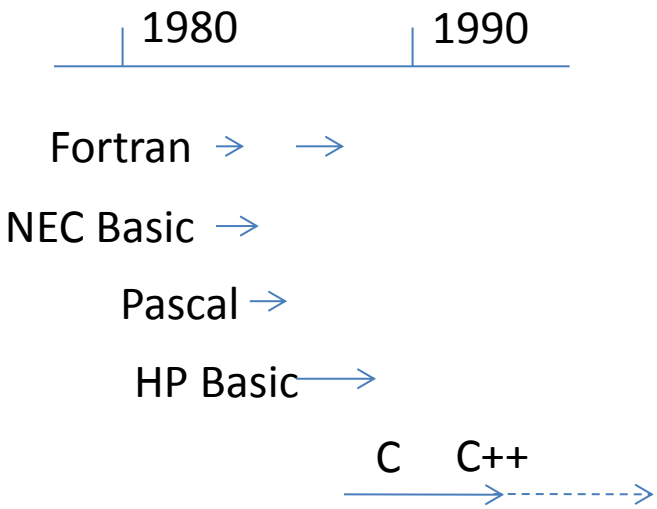


Before CINT (author's experience)

In 1980s,

- New computer language every year as computer evolves
- Had to re-write library each time

Needed stable computer platform!!!



Meanwhile, software guys were using C

C/C++ language is

- Industry standard	good
- Not easy	bad
- Compiler was very slow	bad
- No interpreter	bad

Decided to solve those issues by myself to obtain stable computation environment for >10 years

CINT Original Concept

Sustain scientific computation environment for many years

- Support ANSI/ISO C/C++ — Sustainable
  - As Implementation:
    - Multi Platform, Compiler independent
    - Exclude non-standard components such as graphics
  - As Scripting Language:
    - May not be fully compliant, but should be good enough for scripting
- Easy to Use — Quick & Easy
  - Fast Turn Around Time, Interactive
  - Language short-cut (Complexity of C++ remains as an issue, though)
- Seamless interpreter-compiler integration — Fast
  - Seamless access between interpreter and compiler code
- Portable Component — Flexible
  - Can be integrated with any foreign library or framework

CINT Development

- At beginning, it wasn't a serious project 1990
  - The author had absolutely no C/C++ knowledge
  - Intended to be local tool
  - It worked somehow…
- Adding C++ features 1992
  - Object Oriented Programming (Class, Inheritance, Virtual Function, etc…)
  - Complicated, but not so bad
- Adding STL 1994
  - Generic Programming - Great concept but extremely difficult to implement
  - Template itself is complicated enough, STL is a nightmare on top of it
- ROOT collaboration 1995
  - Add special feature
  - HEP became the largest user community

Challenges

- Complexity of C/C++
  - Template and STL
  - Inheritance, Function overloading, Operator precedence, etc…
- C/C++ as Interpreter language
  - Strongly Typed Language definition conflicts Ease of Use
- Interpreter-Compiler Seamlessness
  - Interpreted class inheriting compiled class
    - (protected member, virtual function, Pointer to member function)
  - Dictionary code was an inevitable solution to realize this concept

Ad-hoc approach turned out to be right for CINT

- Tried to think thoroughly but overwhelmed by challenges

Alternatives were …

- As time goes by….
  Before: Many MPUs, OSs, compilers
  Very slow Compiler/Linker
  Now: Only 2-3 MPU&OS choices - Intel / Win, Linux, Mac / VC++, gcc
  Quick Compiler/Linker
- Platform dependent features may not hurt so much…
  Depending on Compiler/Processor specific undocumented scheme will simplify or eliminate dictionary code
  Data stacking, Pointer to member, Name mangling, …
- Just implement as another compiler
  1. Byte-code compiler + virtual machine
  2. Use compiler/linker in background (ACliC approach)

Some features may be lost, but probably still be usable

The Biggest Road-Block

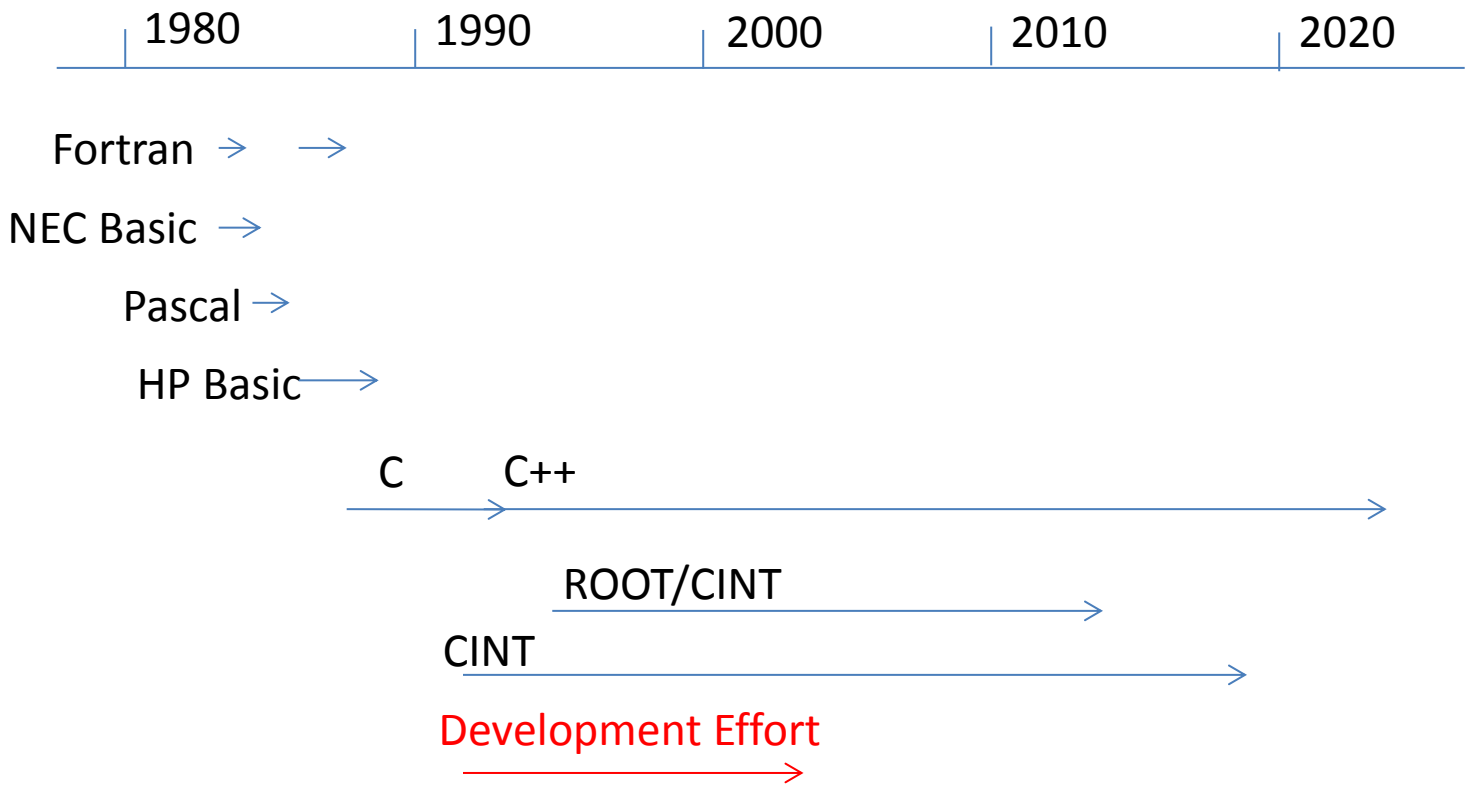
- Spam mail
  - At Iraq War (2003), Cyber attack began
  - Number of Spam mail increased >x10
    - 100s of Spam mails per day
    - Slow internet connection
    - Spam filter were not ready
  - Took 2-3 hours just to download and filter e-mails

Stopped being an active developer but remain as a user

- CINT already had enough capability for my original objective
- -> "*ROOT/CINT used in Test and Measurement Industry*"

Overview and Conclusion

- CINT provided Easy-Interactive-Fast computation environment
- Lasted more than 20 years helping our productivity



Thanks to CERN & Fermi-Lab collaboration, CINT was very successful

Acknowledgement

- Special thanks to
  Rene Brun
  Fons Rademakers
  Philippe Canal
  Axel Naumann
  all ROOT/CINT developers and users