CHEP95 Conference Summary

J. N. Butler

Fermilab

Pros and Cons of Giving The Summary Talk

- You don't have to prepare -- other people really did all the work
- You get the last word

- You can't prepare
- You have to stay awake for the talks
- Probably everyone just wants you to finish so they can relax before the banquet

The Present and Recent Past

- HEP software has met the challenge of a new round of sophisticated experiments at
  - LEP
  - Fermilab
  - Hera
  - CESR
  - Slac
- Data had been collected (DAQ,trigger), recorded, reconstructed, analyzed, and published in a timely fashion.

No Rest for the Weary
The New Challenge

- Field is driven by physics questions which involve either higher mass scales or more precision. This is enabled by:
  - Machine people who continually find innovative ways to increase the energy and luminosity and
  - Detector builders who construct ever-more sophisticated devices with more and more channels

The Gathering Storm
(aka Challenges for the Future)

- HEP Class of (approx) 2000:
  - CLEO III, BaBar, Belle
  - Fermilab Main Injector: D0, CDF
  - Hera-B
- HEP Class of (approx) 2005:
  - LHC: ATLAS, CMS, ALICE, LHC-B
  - Other - TEV33, upgrades

Why the Drama?

- Margin of victory last time was probably small and caused lots of hard work
- Requirements of new experiments are staggering by current standards
- Fear that our methods will not scale - especially the human resources
- Context in which the problem is solved is getting more complicated both technically and sociologically

Outline - I

- Requirements for future experiments
  - esp DAQ and Trigger
- The $\tau_{EXPERIMENT} > \tau_{CHANGE}$ problem
- The dominance of the commercial sector
- operating system issues -- VMS to UNIX? Why Not Tarry?
- Visualization
- What's hot, what's not
  - Not hot -- CPU cycles
  - Hot -- data storage/ DATA ACCESS

Outline II

- Software and Software Management

  A historical, mythological interlude

  Issues:

  - Project organization and management
  - Languages
  - The N^2 and N^2 problem
  - Communications
  - Looking outside (part II)

The $\tau_{EXPERIMENT} > \tau_{CHANGE}$ problem

- The time between significant hardware changes is now maybe 18 months
- The time between significant operating system changes, significant new products, etc is maybe 36 months (??)
- The time between major steps in software engineering has been 30 years (more later) but we can hope

- Danger of choosing too soon
- Danger of waiting too long
- Failure to plan for long haul -- make a PROGRAMMING SYSTEM PRODUCT
- Failure to plan for change in a fundamental way.

Example of importance of time scales

- Consider a total move to C++
  - On the LHC time scale, no one can argue that it can't be done because of training overhead
  - For the HEP2000, it could be a bit tight

Operating System Issues

- This should NOT be a big issue compared for example, to learning a whole new programming model and methodology.
- Main issue at Fermilab turns out to be breaking people loose from a an attitude of complacency that was not justified and stood in the way of progress.

Operating System Issues

- However, the most interesting argument is the Why Not Tarry argument.
- I believe that the commercial world REQUIRES an open operating system and that innovation will be stiffled without it. We will have a UNIX-LIKE thing around for a while.

Operating System Issues

- I do not believe that NT now has the robustness, tools, and appropriate support for HEP production and analysis for the class of 2000. However, Sverre Jarp's efforts seem to say that it's much closer than I think it is ...
- Anyhow, we would be foolish to ignore NT. Fermilab has made NTAS the cornerstone of its personal computing support strategy and...

Operating System Issues

- on the advice of many people, we are cautiously beginning investigations of whether NT can be the basis of certain kinds of desktop analysis. We are also looking into production farms of PC's using LINUX or NT. The Fermi UNIX Environment group has raised the issue of its relation to NT.
- The excellent work of the personal computing companies on Object Oriented techniques, the power and energy and sheer size of the effort on NT means it would be foolish to ignore it. See talk by David Foster.

What's HOT?

- data storage/DATA ACCESS
- Data Storage

  When we studied this for RUN I, disk cost $2.00/megabyte and robotic storage cost $0.50/megabyte. Both have come down quite a bit - maybe $0.30 and $0.10???

  New tape and drive technology is about to push cost to $0.01-- 0.005/megabyte for robotic tape storage.

HOT

New technologies as covered in the talks promise even higher densities and hopefully lower costs in disk. We hope tape costs will drop at least by a factor of 2 more. We can be optimistic that even though are data storage needs are HORRIFIC, we can somehow manage them BUT .....

HOT -- ACCESS TO DATA

- There were many talks on this topic. The best symbol I can think of is the final scene from Raiders of the Lost Ark.

  We have eliminated the analysis bottlenecks right up to the final analysis, often done on the physicists desktop.

- Access includes

  finding data by invoking a logical namerather physical description of the file

  Getting it back without any explicit interactoin with hardware and in a timely fashion (high speed -- not horrible latency)

HOT -- ACCESS TO DATA

- Some basic issues
  - Bandwidth
  - hardware organization -- CPUs, disks, transport technology
  - Data organization -- including HSM, locating of data, etc-- and data retrieval
  - Query Language
  - Nile, CAP, PASS, National Scalable Cluster Project, and more...

HOT - ACCESS TO DATA

- General consensus that sequential file access not the answer. Some kind of special data organization and access via database or object database technology seen as the answer.

What's not

- CPU cycles for event reconstruction and Monte Carlo

  When we made our last farm buy for RUN I -- CIRCA 1992 -- a "MIP" cost $140.

  A "MIP" probably now costs $30-75.

  30,000 MIPS for D0 or CDF event reconstruction will cost about $1 million. Compare to $30-40 MILLION FOR DETECTOR UPGRADE.

Not Hot

- the price will probably go down by another factor of 2-3 before we have to commit for RUN II, which is good because we haven't included efficiency, reprocessing, or Monte Carlo.
- Lowest price will certainly be in PC mass market machines. Workstations will be more expensive

Not Hot

- Big discussion -- do more integrated solutions like SMP machines, which cost more, really win over loosely coupled machines if you put in the efficiency factors. DESY or CDF/FNAL Challenges vs FNAL farm or distributed clusters.
- People should still try to write reasonably efficient code. Also, watch out for memory use or memory costs can dominate

A Historical and Mythological Tour

- Knowing that software engineering would be a big issue, I consulted the first book I ever read on the subject:
- The Mythological Man-Month, Frederick P. Brooks, Jr., 1975
  - Father of IBM System/360
  - Manager of OS/360 in 1964 and 1965
  - Book reprinted with corrections in 1982
  - Recent update

Software Projects -- Brooks

- Task Profile
  - 1/3 planning
  - 1/6 coding
  - 1/4 component test and early systems test
  - 1/4 system test, all components in hand
- Note that half of the schedule is devoted to debugging and optimizing
- Coding, which draws much attention and is easiest to estimate, is only 1/6 of task

Communications -- Brooks

- Do all things possible to reduce the interactions of the parts of the program
  - Achieve conceptual integrity oin design
  - Design must proceed from one or a very few agreeing resonant minds
  - Subnet very large projects to achieve this and to reduce N- effective.

Communications -- II

- Use all possible tools to improve communication -- specifications, formal definitions, conferences, workbooks
- Choose tools based in large part on reducing the complexity of interactions
- Plan the organization for change

Epilogue
The Mythical Man Month

The tar pit of software engineering will continue to be sticky for a long time to come. One can expect the human race to continue attempting systems just within or just beyond our reach; and software systems are perhaps the most intricate and complex of man's handiworks. The management of this complex craft will demand our best use of new languages and systems, our best adaptation of proven engineering management methods, liberal doses of common sense, and a God-given humility to recognize our fallibility and limitations.