Event Processing Frameworks
a Social and Technical Challenge

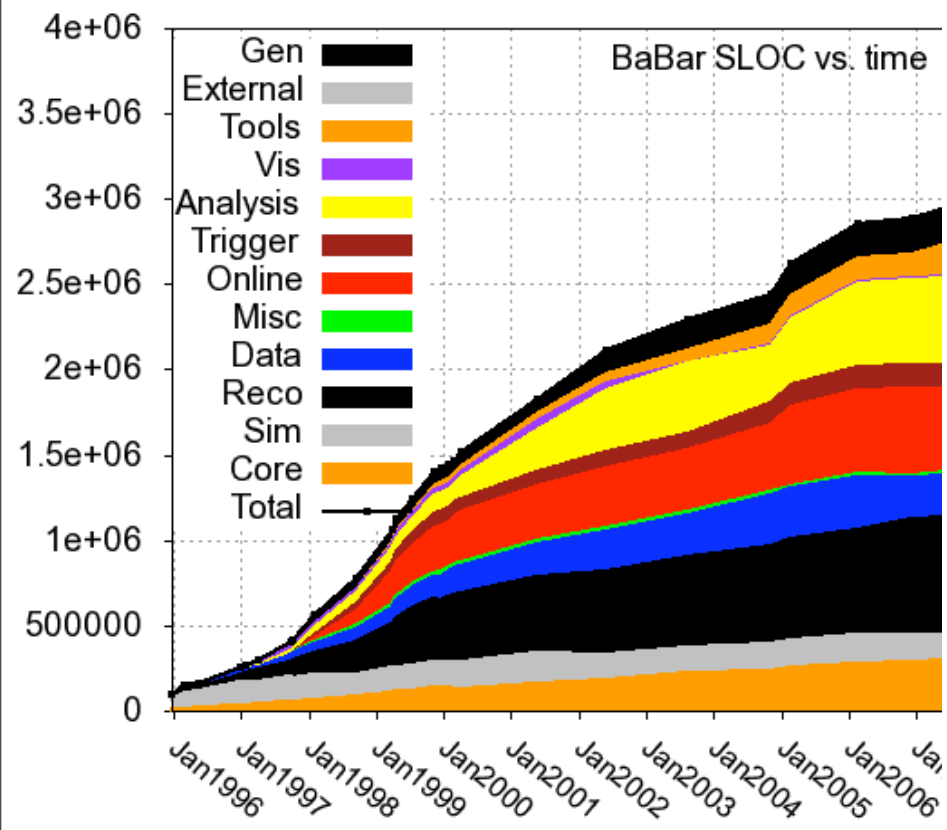
Liz Sexton-Kennedy

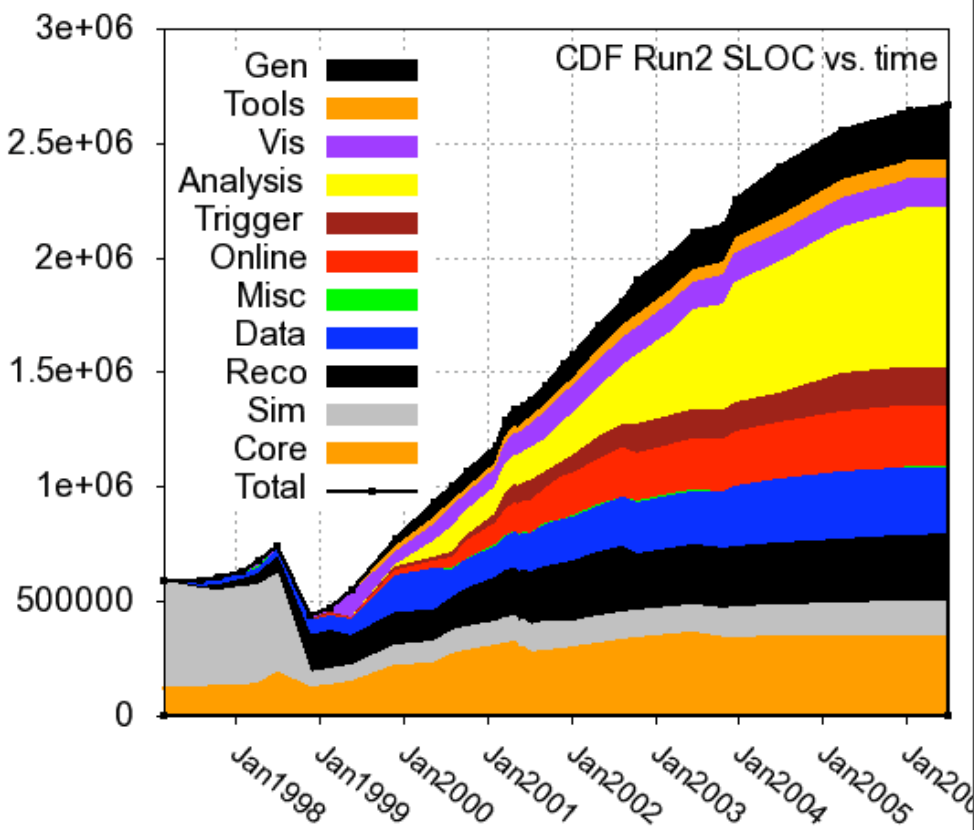
My Background

- Student on fixed target experiments ’82-’86
- Started working for the CDF online group in 1988. I was hired to support their Fortran framework.
- Developer of the L3 trigger for CDF run 1
- Project engineer for the offline software upgrade for run 2. The switch to C++ started in 1996.
- Chose to collaborate with BaBar and CLEO
- Lead run 2 developers, eventually became head of offline software and computing, ‘96 – ’04
- Joined CMS in ’05, lead the Fw/Edm reengineering project.

What is the Problem?

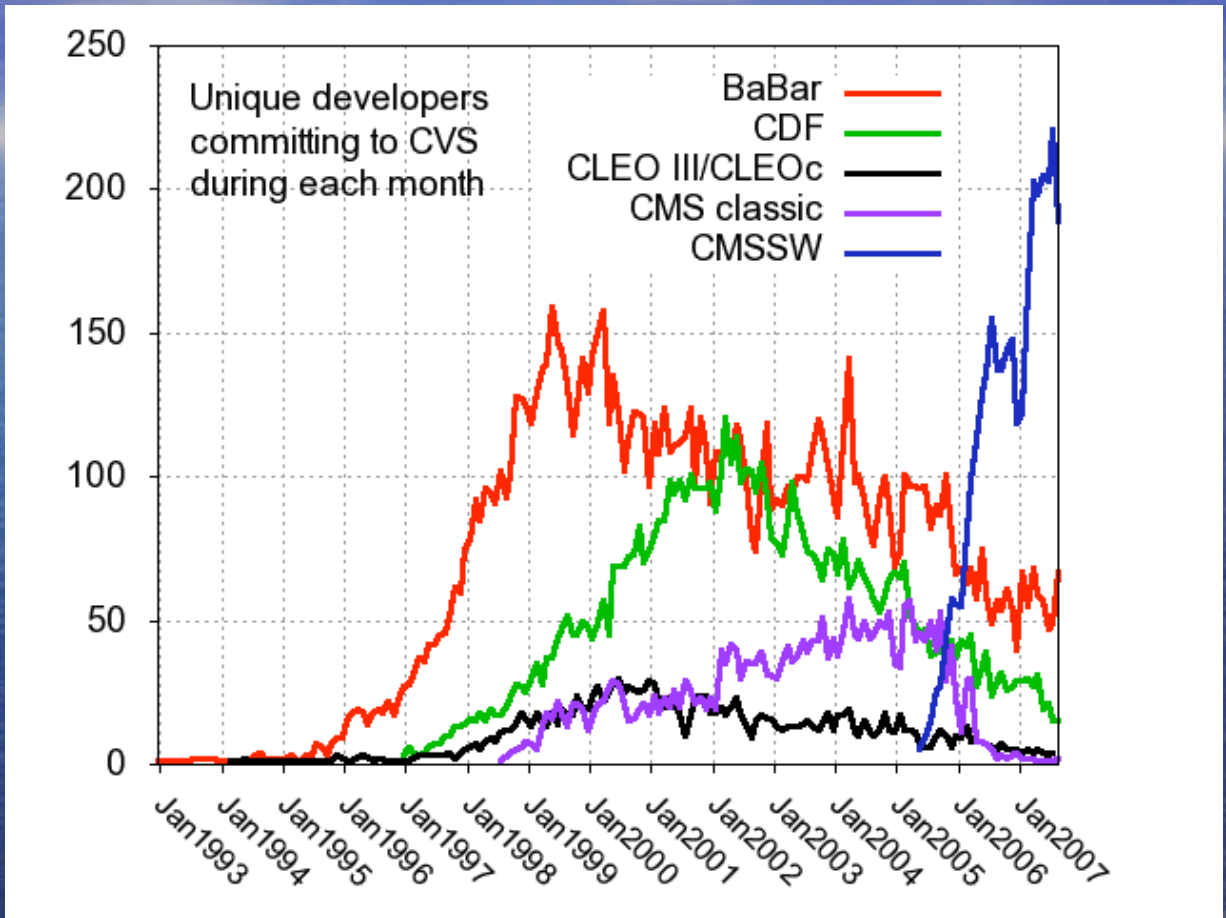
- HEP event reconstruction applications are large scale projects, requiring millions of lines of code.
- This is the technical challenge.





What is the Problem?

- HEP event reconstruction applications need to integrate the creative output of 10s to 100s of part time physicist programmers.
- This is the social challenge.

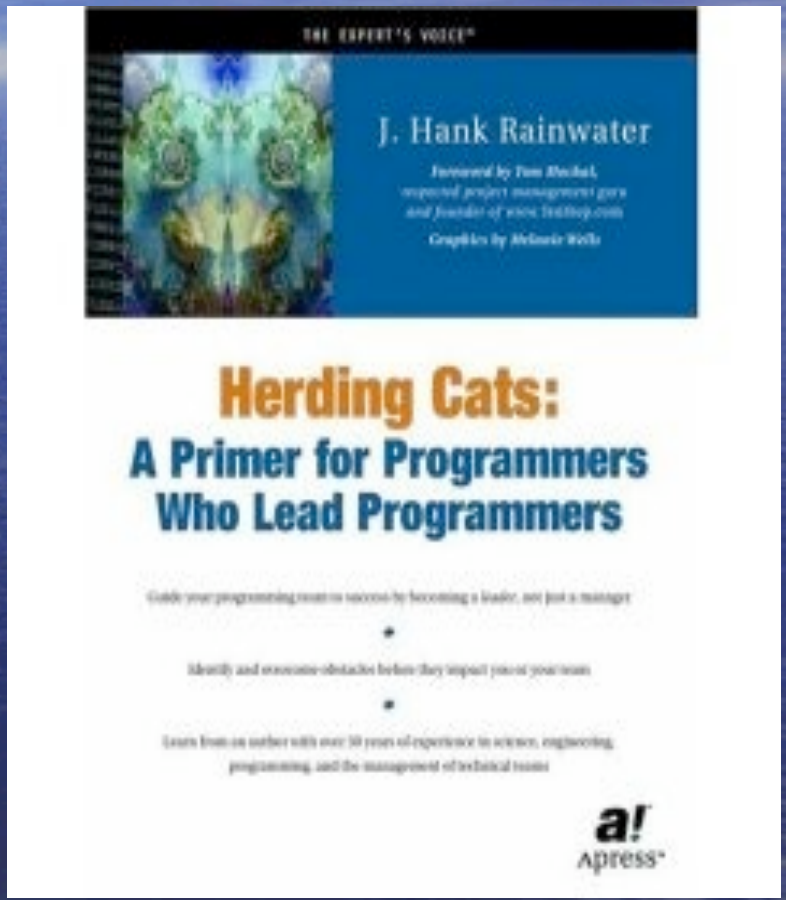


Frameworks Provide the Means of Organization

- If successful the Framework/EDM (Event Data Model) is the one project that all developers and physicist need to know the design goals, rules, and API's of.
- It's not successful if it's not widely used.
- Technical as well as social factors determine how much it is adopted.
- It must be easy to learn, yet functional enough to make it worth using.
- Developers often vote with their feet…

Herding Cats

- The type of people that use these systems are very independent and creative.
- The type of people who design them are few.
- You have to be able to explain to them why using your system will make their task easier.
- You must limit the complexity, code elegance is not the primary value to them.
- The pattern set in the reconstruction will be reused for analysis if successful.



Frameworks Manage the Required Technologically Complex Software Systems

- These include:
  - Event data persistence
  - Relational DB interfaces
  - Other "services"
  - Provenance or history tracking
  - Configuration
  - Workflow scheduling
- It must be able to scale to applications with many millions of lines of code, and remain operable and debuggable.

Past Frameworks in Retrospect

- In the past (’83-’96) Zebra/YBos banks formed a white-board EDM; familiar Fortran concept reused.
- Usually some concept of “user analysis” component architecture.
- Some, e.g., CDF’s framework even used a software bus processing model in Fortran.
- These frameworks were widely used in triggering and primary reconstruction but not in the analysis phase … they used PAW
- CDF repeated the pattern for run 2; AC++, root
- The next slide is a feature table for these first generation C++ experiments.

Feature Survey for (Recently) Running Experiments

	EDM? And persistency	Configure?	Processing Model?	**Component Architecture?**	Provenance?
BaBar	Transient -> OODB	With tcl	Software bus	Yes but static link	Parameters captured DB
CDF	Transient -> root Ttrees	With tcl	Software bus	Yes but static link	Parameters captured DB
D0	Transient -> DOOM	With text rcp files	publish / subscribe	Yes and dynamic link	Saved to data file
CLEO	Transient -> custom	With tcl	Data on demand	Yes and dynamic load	Limited
Belle	Transient -> custom	With script parser custom	Software bus	Yes and dynamic load + mixture	Stored in log files
STAR	Transient -> Many-tables			Yes	

How Well Did These Work?

- As measured by the level of adoption and scalability were these successful?
- Only CLEO reported a great success in the '03 CHEP conference
- CDF and D0's infrastructure was successful in L3 and primary reconstruction, however all secondary DS and analysis were done with Ntuples … so a partial success.
- BaBar had to redesign their persistency model after 3 years of running.

Frameworks for LHC Era

	EDM? And persistency	Configure?	Processing Model?	**Component Architecture?**	Provenance ?
LHCb	Transient -> pool	Text file -> python	Data on demand	Yes and dynamic link	Config. Based store in DM DB
Atlas	Transient -> pool	python	Software bus	Yes and dynamic link	Config. Based store in DM DB
CMS	root Ttrees	With text -> python	Software bus & Data on demand	Yes, uses edm::Plugin Manager	Stored in output data & DM DB + per event info
Alice	root Ttrees	cint	Software bus	Yes, uses root::Plugin Manager	Config. Based store in output

How Well Are These Working?

- Adoption levels within the experiments is high. The scale of the problem has grown a great deal over the years so the "role your own" method is not an option.
- Scaling: without data it's hard to tell. So far LHC MC exercises have exceeded the scale of the run 2 experiments.
- It will be interesting to see if CMS's focus on provenance tracking gives it a competitive advantage.

For More Technical Information

- CHEP09 conference – Event Processing: Monday
- CHEP06 - relevant session
  - indico.cern.ch/sessionDisplay.py?sessionId=3&slotId=0&confId=048#2006-02-13
    - Look for CMS papers by Chris Jones

Conclusions

- Frameworks should be judged by how well they meet the technical challenges of complexity and scalability.
- However that is not enough, it must also be an easy to use set of tools that organizes large groups of developer physicists into moving in the same direction