Applying Object-Oriented Software Engineering at BaBar

A summary of

¥our approach to engineering object-oriented reconstruction software

¥practical experience, successes and failures

¥where we re going

Outline:

¥Design experience

¥Results to date

¥Where next

Disclaimers:

¥I do event reconstruction, the primary OO part of BaBar so far. Other parts of the overall computing system are evolving in different ways.

¥These are my own opinions, not necessarily those of BaBar leadership.

Related Talks at CHEP97

Architecture of the BaBar Reconstruction System: Practical Issues

(Tuesday 10:30 P5)

Panel Discussion: Transition to OO?

(Wednesday AM)

The BaBar distributed development / centralized release system

(Wednesday 18:00 P3)

Applying Object-Oriented Software Engineering at the BaBar Collaboration

(Thursday 14:00 P3)

An object-oriented tracking system for BaBar

(Thursday 14:30 P5)

An object-oriented extended Kalman filter tracking algorithm

(Thursday 16:00 P5)

Beta: A high level toolkit for physics analysis

(Thursday 17:00 P5)

The BaBar experiment

BaBar is the detector at the SLAC PEP-II BFactory

Running starts in 1999:

- ¥2kHz level 1 trigger, 50MB/sec to level 3 trigger farm
- ¥About 100 Hz of recorded events, 3MB/sec, 10^9 events/year
- ¥Running >> 80% of the year for >> 5 years

Semi-conventional detector, but new level of precision

- ¥5 layer vertex detector & low-mass driftchamber
- ¥Crystal calorimeter & finely segmented iron flux return
- ¥Cherenkov particle ID (DIRC)

Big, distributed collaboration

- ¥80 institutions, >500 people
- ¥All over the world

The detector must be built from scratch; should the software also?

Reconstruction Goals

Usable reconstruction when the first data arrives (1999)

- ¥Complete reconstruction suitable for physics
- ¥Alignment and constants-finding ready for data
- ¥Data quality checks in place
- ¥Sufficient performance available

Maintainable, extendable system

- ¥Must survive contact with real data, physics
- ¥High quality operation under realistic conditions within first year
- ¥Must be maintainable for > 10 years
  - Frequent extensions
  - Difficult conditions

Not Reconstruction Goals

Advancement of the state of the software art

¥ Really cool CHEP97 papers

Support for short term studies

¥Detector design studies

¥Today s reconstruction algorithms

Ò Purveyors of software to the HEP communityÓ

¥Cooperation, not evangelism

These things are:

¥Needed within BaBar, and in larger community

¥Important

¥Interesting

¥Sometimes necessary for us to do

¥But also very distracting

Reconstruction effort must stay focused

Reconstruction Principles

Factory running requires a high degree of quality control.

¥Frequent reprocessing is unaffordable luxury

Reconstruction should behave identically when doing identical things.

¥Not just all reconstruction jobs built from same source code

¥Production run, analysis run give same results

You can rerun a reconstruction job and obtain the same results.

¥Distribution in both time and space

The results of reconstructing an event must not depend on which other events have been processed.

¥Events are independent

¥Must be reconstructed independently.

Not all events will be processed the same.

¥Bhabha events different from hadronic events, etc.

Multi-prong attack

Architecture

- ¥Solving the too many first choices problem
- ¥OO design expressing traditional concepts

Iterative design & implementation process

- ¥Applying evolutionary pressure
  - Design and implementation intertwined
- ¥Strike multiple balances
  - Learn by doing, do while learning
  - Getting something in place vs. design work
- ¥In spite of imposed waterfall-model schedule

Gain control of the process by controlling the product

- ¥Code management
- ¥Quality control and assurance

Team-building

- ¥Getting the people
- ¥Formal training balancing experience and exposure

Iterative design process - Tracking as example

Starts slow, builds up detail

¥Good match to rising skill, experience levels

Done with a combination of:

¥Rose, drawing tools

¥Writing code and testing it

¥ Booch method

Spread across world

¥Phone meetings

¥Web pages animated by HyperNews

¥Many with opinions, four or five working

Very hard to get people to work on collaborative design effort

Three types of reconstruction code

Modules

¥Definition: Coherent chunk of software to do specific input -> output

¥We re trained to think in terms of dataflow

data not the same as objects

well mapped to BOS, ZEBRA, etc

¥ Module is good way to build complicated results

Three different track finding modules can coexist

Track finder

Objects

¥Objects encapsulate complex behavior yet to happen

Module is much more static

Object much better suited when not enough known yet

Track

The surrounding support environment

¥Needed to construct and control a complex reconstruction job

Reco Note 1, Structure of the BaBar Reconstruction Software at
http://www.slac.stanford.edu/BF/doc/Computing/Reconstruction/Notes/RecoSoftStructure.ps

(insert a bunch of slides here)

Ideas always start out rather naive:

http://www.slac.stanford.edu/BF/doc/Computing/Reconstruction/Tracking/trk1/Rose_temp.ps

Understanding comes from experience, running code is a great source of experience

¥Event display picture, showing hard scatter from wire, DCH tracks that don t match

Design is keeping pace with understanding

http://design.lbl.gov:80/~brownd/new_tracking_classes.ps

¥Note ad-hoc selection of tools

http://www.slac.stanford.edu/BFROOT/doc/Computing/Reconstruction/Tracking/RecoTrack.ps

Open communications, by holding hypernews discussions:

http://www.slac.stanford.edu/BFROOT/doc/Computing/Reconstruction/CHEP97/applyingOO/TrackingHN.ps

¥Note that many of these are large and detailed

http://www.slac.stanford.edu/BFROOT/doc/Computing/Reconstruction/CHEP97/applyingOO/TrkOneItem.ps

¥Note emphasis on discussing design of implementation, and what the next step is

We start showing plots of various reconstructed quantities

¥particularly useful to compare multiple algorithms/designs at early stage

Slowly, the design grows and merges

¥geometry & tracking interface in a Kalman filter

http://babar-hn.slac.stanford.edu:5090/BFROOT/doc/Computing/Reconstruction/Tracking/kalman.ps

¥Progress driven by understanding, perceived needs of the work

Lots more disagrams of the geometry system

Drawn pictures of it working

¥For Kalman

¥For the Cherenov PID, a second system

Also have workshops to bring people together

¥By the end, the agenda becomes open discussion

http://www.slac.stanford.edu/BFROOT/doc/www/CompRecoWkshpFeb97.html

Example slides (not reproduced, as handwritten AT the workshop)

¥Use case for our muon system

¥You can actually see the design process take a loop there

¥Its important to get people doing this

How do you measure progress quantitatively?

First, count lines of code, number of files

¥Have to start somewhere

Getting more sophisticated

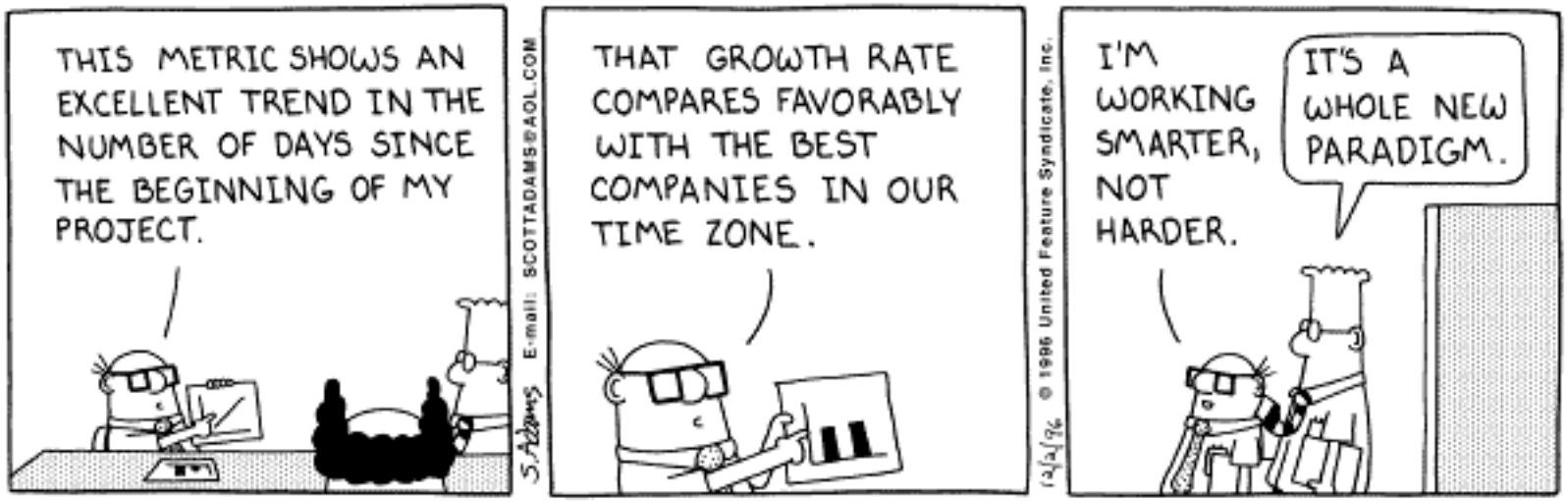
¥Package dependencies, class sizes

¥R. C. Martin s metrics toolkit

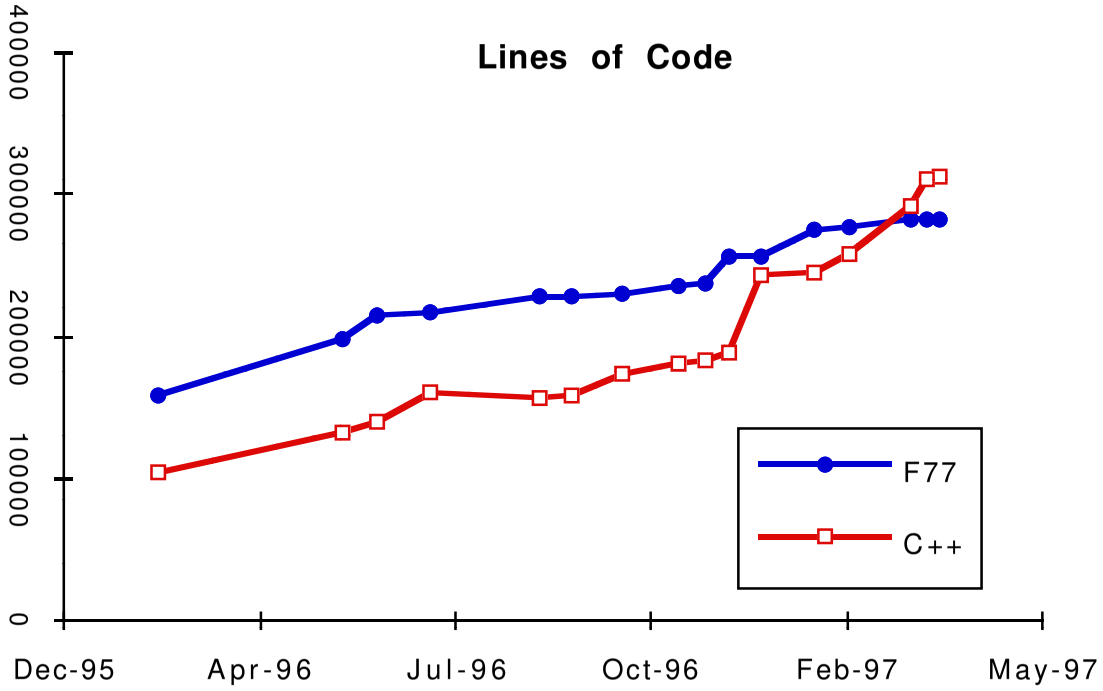
¥Number of defects found in the release process

¥GNATs to record bug reports, time to repair

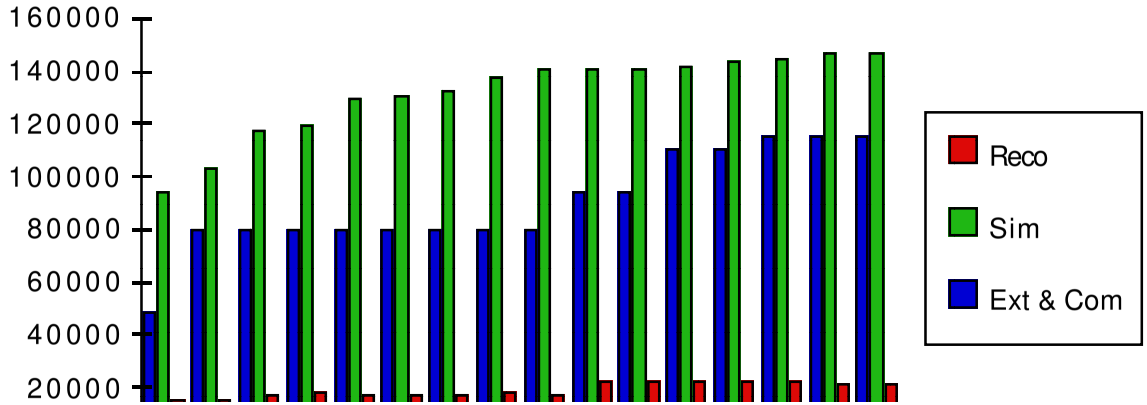
Making progress, but a long struggle



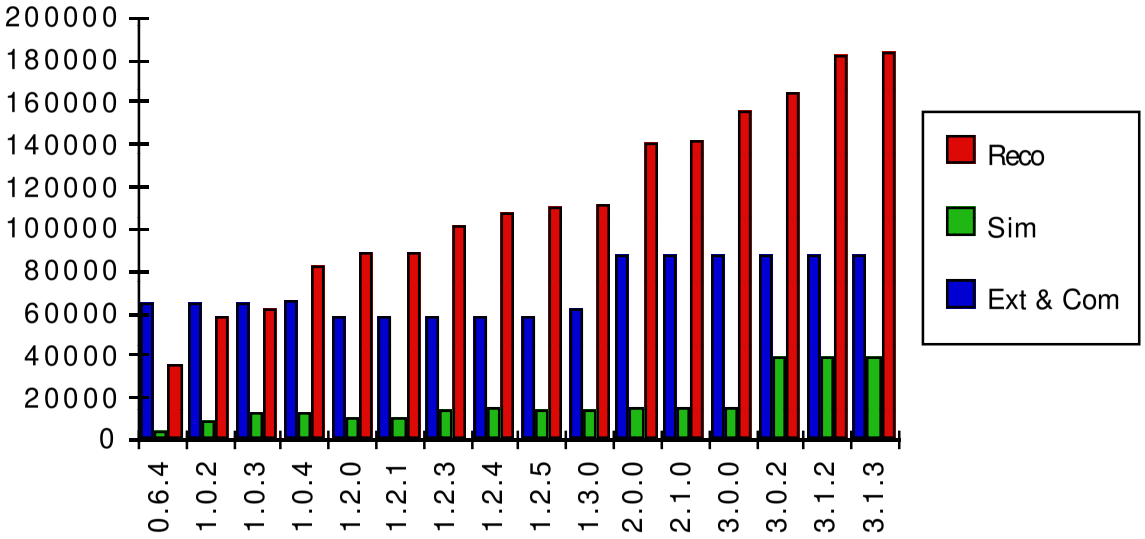
Copyright © 1996 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited


Lines of Code
400000
300000
200000
100000
0
Dec-95
Apr-96
Jul-96
Oct-96
Feb-97
May-97
F77
C++

Fortran77 Lines of Code



C++ Lines of Code



Analysis Connection

Analysis is our customer

¥Since customer is always right, have to provide a product they can use

Two pronged approach:

¥Provide a simple, extensible toolkit

Allows glass-box access to reconstruction output

But lets user do simple stuff in a simple way

¥Get as many people as possible up to speed on the native output

Teach technicalities, explain design structure, etc.

Primarily through visitors, etc, from the new generation

These people are also implementing that toolkit

To be usable, the toolkit has to use concepts for non-experts

¥Makes it look a lot like super-FORTRAN, but that s OK

¥At least 13 people have used it

to at least the level of sending in bug reports and questions

Biggest need here: people to support each other, build a user community

Coming Events - The First Mock Data Challenge

Multiple reasons

¥Kick-start the physics effort

¥Spread knowledge of our sim & reco software

¥...

But primarily a development tool

¥ You can t tell how well its working until somebody else tries to use it

Six month effort

¥smooth out the rough edges

¥build the missing pieces

Copyright © 1996 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Summary

Applying modern software engineering is a long, slow process

- ¥C++ is a step in that direction
- ¥Engineering discipline not yet routine

Trying to build a system in the larger sense:

- ¥Skilled people
- ¥Habit & Procedures
- ¥Infrastructure
- ¥Designs and documentation
- ¥Code

This requires solving several problems at once:

- ¥Build the infrastructure the effort needs
- ¥Bring the people inside and outside the computing system along
  - They have to see that its worthwhile
  - We have to make it worthwhile
  - And it will still be slow
- ¥Produce the actual reconstruction code

OOA/D/P has promised impressive things, and we intend to deliver