Migration of reconstruction and analysis software to C++

A proposal based on feedback from the software week
Immediate Goals

◆ **Physics Goal**:  
  - To be able to run new tracking pattern recognition algorithms written in C++ in production with standard FORTRAN algorithms in time to produce useful results for the RICH TDR.

◆ **Software Goal**:  
  - To allow software developers to become familiar with GAUDI and to encourage the development of new software algorithms in C++.
Proposed Strategy - Step 1

Finish splitting of SICB into:

- simulation (SICBMC)
  - Event generation, GEANT tracking
  - outputs kinematics AND raw GEANT hits
    - i.e. Current RAW format, but with xxRW banks from all detectors

- reconstruction (SICBREC)
  - doesn’t need GEANT3 nor its common blocks
  - digitisation, trigger, reconstruction in distinct steps
  - outputs same DST format as now.

Benefits:

- Clear separation between simulation and reconstruction
- Modularity of reconstruction

Organiser: Florence
Step 1 overview

SICBSIM

Generator

MC Kinematics

GEANT transport

SICB “raw” data

Entrance and exit points and energy loss in detectors

SICBREC

SICB local tracking and “digitisation”

SICB “digitisings” (not on SICB DST)

Trigger

Pattern Recognition

Half way between raw data and coordinates. E.g. wire number, signed drift distance

SICB DST

Marco Cattaneo, 1st December 1999
Tasks for step 1

- **SICBMC**
  - Remove anything that belongs to digitisation and reconstruction
    - Done
  - Create raw hits for calorimeter
    - To be done by calorimeter experts (~1 week?)

- **SICBREC**
  - Add initialization routines for each step of the processing (digitize, apply trigger, reconstruction)
  - Verify validity of results
    - Both the above are essentially work for the sub-detector experts
    - Time estimate is about two weeks

- If started now, step 1 could be finished by Xmas

Marco Cattaneo, 1st December 1999
Step 1
SICBREC structure

SICB raw

Digitisation A
Digitisation B
Trigger
Reconstruct A
Reconstruct B

This dataflow is forbidden (no COMMON...)

SICB DST

SICB “framework"
Proposed Strategy - Step 2

- **For every SICBREC FORTRAN module:**
  - wrap it such that it can be called from C++
  - Integrate with the GAUDI framework
    - Time estimate about 1 month - sub-detectors and Gaudi team
  - Result is a new reconstruction program - BRUNEL

- **Produce a DST (Zebra banks) with this program**
  - check the output is as expected
    - i.e. identical to SICBREC output
    - Must be done by sub-detector experts

- **Drop SICBREC**
  - Could be ready for decision by next LHCb week

- **Benefit:**
  - Single environment for C++ and FORTRAN work
  - Integrated environment for verification of C++ developments

- **Organiser:** Marco
Step 2
BRUNEL structure

SICB raw

Input

Digitisation A

Digitisation B

Trigger

Reconstruct A

Reconstruct B

ZEBRA common (SICB banks)

Gaudi framework

Output

SICB DST

Marco Cattaneo, 1st December 1999
Proposed Strategy - Step 3

- Start replacing FORTRAN modules with C++ equivalent. Each new piece consists of:
  - event model
  - detector description
  - algorithm.

- Provide converters to:
  - regenerate same SICB output bank that was there before
    - Preserves format of SICB DST
    - DST banks may contain improved data (e.g. Result of tracking pattern recognition)
    - Some “added value” of C++ algorithms would NOT be on SICB DST
  - write data out to the supported persistent object manager
    - Contains as complete a reconstructed event as is available in GAUDI event store
    - Including “added value” of C++ algorithms, available only to GAUDI based analyses
Step 3 - organisation

- This step implies (for each sub-detector):
  - Development of event model: help coordinated by Marco
  - Development of detector description: help coordinated by Florence
  - Reviews of evt mod, det desc, algorithms: organised by John.
    - Review panel will include Per, Marco, Florence, plus SDs

- Known candidates:
  - Tracking,
  - Analysis tools,
  - Muon digitisation,
  - Calorimeters,
  - RICH,
  - ...

- Timescale:
  - Depends (almost) entirely on sub-detectors
Step 3
BRUNEL structure

SICB raw

Digitisation A
Digitisation B
Trigger
Reconstruct B

CDF files

ZEBRA common (SICB banks)

Sicb Converters
Sicb Converter

Gaudi transient event store

BRUNEL

XML geom.

Reconstruct A

OO DST

SICB DST

Zebra Input

Zebra Output

OO DST
Step 3: Analysis structure

Analysis A

Has access ONLY to SICB DST
Steered by GAUDI job options

ZEBRA common (SICB banks)

Sicb Converters

Have access to OO DST and SICB banks for which converters exist.
Steered by GAUDI job options

GAUDI

OO DST

Zebra Input

SICB DST

HBOOK

GAUDI job options

OO Input

Input

Analysis B

Visualisation
Step 3 - benefits

- A unified development and production environment
  - As soon as C++ algorithms are proven to do the right thing, they can be brought into production in the official reconstruction program

- Early exposure of all developers to Gaudi infrastructure
  - FORTRAN gurus and C++ beginners

- Increasing functionality of OO 'DST'
  - As more and more of the event data becomes available in Gaudi, it will become more and more attractive to perform analysis with Gaudi
  - N.B. Contains ALL (and only!) parts of reconstructed event for which data model is defined

- A smooth transition to a C++ only reconstruction
Summary

- **Step 1:** separate SICBMC and SICBREC
  - Could be ready by Xmas

- **Step 2:** wrap SICBREC algorithms into Gaudi framework
  - Could be ready by end February

- **Step 3:** gradually replace FORTRAN with C++ algorithms
  - Timescale dictated by sub-detector priorities
  - Development/integration in Gaudi can start now

- **Analysis in Gaudi is possible now**
  - Functionality will increase as subdetectors define their data model
  - Analysis toolkit under development, send requirements to Gloria

Marco Cattaneo, 1st December 1999