Software for the LHCb Experiment

G. Corti, CERN/PH
for the LHCb Collaboration
The LHCb experiment

- Designed to make precision measurement of CP violation and other rare phenomena in the b system at the LHC
- Trigger and reconstruct many different B decay modes to make independent and complementary measurements

LHCb is a single arm forward spectrometer

Forward production of $\bar{b}b$, correlated

$12 \text{ mrad} < \theta < 300 \ (250) \text{ mrad}$

i.e. $2.0 < \eta < 4.9$

N21–8, W. Witzeling
“Status of the LHCb experiment”
LHCb Software Strategy

- Develop an *Architecture* ('blueprint') and a *Framework* (real code) to be used at all stages of LHCb data processing
  - high level triggers, simulation, reconstruction, analysis
  - a single framework used by all members of the collaboration

- **Avoid fragmentation and duplication of computing efforts**
  - common vocabulary, better understanding of the system
  - better specifications of what needs to be done
  - identify and build common components

- Transparent use of third-party components wherever possible or necessary
  - GUI, persistency, simulation....

- **Applications** are developed by customizing the Framework: *Gaudi*
Software Organization

Applications built on top of frameworks and implementing the physics algorithms specific to the different LHCb software tasks.

One framework for basic services + various specialized frameworks: detector description, visualization, persistency (POOL), interactivity, simulation (GEANT4), etc.

A series of basic libraries widely used: STL, CLHEP, GSL, BOOST, etc.

Frameworks Toolkits

High level triggers

Reconstruction

Simulation

Analysis

Foundation Libraries
GAUDI Architecture and framework

- Separation between “data” and “algorithms”
- Separation between “transient” and “persistent” representations of data
- “Physicist code” encapsulated in few specific places (*Algorithms, Tools*)
- Well defined component “interfaces”
Gaudi Core Services

- Independent components have been developed in incremental steps of functionality and complexity
- General Services
  - ex.: JobOptions Service, Message Service, Event Data Service, Histogram Service, Random Number Generator, ...
- Specialized Services with interfaces to “external” software
  - Data Dictionary Services
  - Event Model Support classes
  - Detector Descriptions and Conditions
  - Object Persistency
  - Data Management
  - Simulation and Analysis
  - User interaction
Event Model

- Coherent definition of LHCb public event data classes for all processing phases
  - sharing and updating of data between sub-detector algorithms
- Guidelines for common approaches to event data model
  - e.g. access to contained objects, access to Monte Carlo Truth
- Utility classes and services are provided
  - Event data service, container classes, references, associators
- Event model described with a high level object description language
  - automatic generation of header files and dictionary

Information

- GenParticles
- MCParticles
- MCHits
- MCDeposits
- MCDigits
- Digits (Raw data)
- MC Data
- Real Data

MC Data

Comparison

Real Data

Processing

| simulation | reconstruction | analysis |

Particles

ProtoParticles

Tracks

Clusters

Digits

MC Digits

MC Deposits

MC Hits

MC Particles

Gen Particles

Simulation

Reconstruction

Analysis

October 21, 2004

ROME2004 - IEEE NSS

Software for the LHCb Experiment
Applications and data flow

Typical phases of Particle Physics software processing are “encapsulated” in various LHCb applications.

Event model / Physics event model

- Simulation Gauss
- Digitization Boole
- Reconstruction Brunel
- Analysis DaVinci
- Gaudi

Shared between all applications

- GenParts
- MCHits
- Detector Description
- RawData
- Conditions Database
- StrippedDST
- AOD
- MCHits
- Digits
- DST

Common framework for all applications

Typical phases of Particle Physics software processing are “encapsulated” in various LHCb applications.
The data processing applications

- **Gauss** - the simulation application
  - generation of proton-proton collisions
  - tracking of particles in the detector and interactions with the material
  - production of "hits" when particles cross sensitive detectors

- **Boole** - the digitization application
  - simulation of detector response
  - transformation of Monte Carlo hits in digitized data in DAQ-buffer format

- **Brunel** - the reconstruction application
  - complete pattern recognition
  - sub-detectors and combined reconstruction: tracks, calorimeter clusters, RICH, MUON, electromagnetic particle identification

- **DaVinci** - the analysis framework
  - supports selections of events and analysis
  - manipulation and analysis tools for general use
  - physicists interact with "particles" and "vertices"
The simulation application: Gauss

Gauss mimics what will happen in the spectrometer to understand experimental conditions and performance.

Event Generation
- primary event generator
- specialized decay package
- pile-up generation

Detector Simulation
- geometry of the detector (LHCb $\rightarrow$ Geant4)
- tracking through materials (Geant4)
- hit creation and MC truth information (Geant4 $\rightarrow$ LHCb)
Software Triggers

Software triggers to run on LHCb online farm

- No access to permanent store. Re-implement I/O Services and interface JobOptions Service and Message Service to Experiment Control System
- Algorithms are developed in the Analysis Application
Visualization: PANORAMIX

- The Graphical display of Detector Geometry and Event Data objects is provided by Panoramix
  - Data from files or produced on the fly
  - Subset of the event data have currently graphical rendering
- Based on specialized visualization and conversion services
  - OnX and Open Inventor
- Scripting based on Python
- Can work with the data processing applications
Examples of Panoramix display
Conclusions

- Applications for all phases of data processing have been developed and are used for massive data production
  - Produced ~ 210 M events with Gauss, Boole and Brunel in 2004
  - Filtering production of selected events starting this month with DaVinci

- The applications are used by the whole collaboration in their every day work

- Common framework and services facilitate migration of algorithms between the different applications

- The framework and the applications are under continuous evolution and new functionalities are provided as necessary