Detector Description in LHCb

Detector Description Workshop
13 June 2002
S. Ponce, P. Mato / CERN
Contents

- Architecture
- Transient layer
- Persistency layer : XML
- Condition Database
- Visualization
- Interfacing Geant4
- Status & examples
Architecture Overview

XML → Transient Store → Visualization (Panoramix)
Condition Database → Transient Store → Simulation (Gauss/Geant4)
GDML? → Transient Store → Reconstruction (Brunel)

Transient Store:
- Calibration
- Materials
- Structure
- Geometry

DAQ?

Analysis (Da Vinci)

13 June 2002 LHCb Detector Description
Architecture

- Sub-Architecture of Gaudi
  - Same principles
  - Transient/Persistent representations
- Focus on the “Physics Algorithm”
- Coherent access to “all” detector data
  - Geometry, Calibration, Slow Control, etc.
Logical Structure

- The basic object is a Detector Element
  - Identification
  - Navigation (tree-like)
- DetElement as information center
  - Be able to answer any detector related question
    » E.g. global position of strip#, temperature of detector, absolute channel gain, etc.
  - Placeholder for specific code
    » The specific answers will be coded by "Physicists"
Algorithm Accessing Detector Data

- \textit{DetectorService}:
  - \textit{beginEvent}:
    - Manages store
    - Synchronization updates

- \textit{Algorithm}:
  - \textit{DetectorData}:
    - \textit{IDetElement}:
      - \textit{IGeometryInfo}:
        - \textit{GeometryInfo}:
          - \textit{ICalibration}:
            - \textit{Calibration}:
              - \textit{IReadOut}:
                - \textit{ReadOut}:

- \textit{PersistencyService}:
  - \textit{ConversionService}:
    - \textit{ConditionsDB}:
    - \textit{Geometry}:
    - \textit{OtherDBs}:

- \textit{TransientDetectorStore}:
  - \textit{MuonStation}:

13 June 2002  LHCb Detector Description  6
Persistency based on XML files

- XML is used as persistent representation of the Structure, Geometry and Materials (eventually also Conditions)
- Mapping each C++ class into an XML element
  - Inheritance emulation (Generic and Specific Detector Element)
  - Relationships using “Links” and symbolic names
- Allow math expressions with parameters and physical units
  - Using expression evaluator (available in CLHEP)
XML Files

◆ Separated XML files
  - By sub-detector and data type (structure, geometry, material)
  - Low coupling of developments
◆ Links between files through references
  - allows to see the whole description as a single XML tree
◆ Versioning done using CVS
◆ Possible migration to the “Conditions DB”
XML Converters

- Capable of converting (one way for the time being) XML into C++ objects
  - Using DOM interface (Xerces-C)
  - Specific converters for specific "DetElement" (to be provided by users)

- Available Converters
  - Structure: Catalog, DetElement
  - Geometry: LVolume, Surface, Solids (various shapes, boolean), PVolumes (parametric)
  - Materials: Isotope, Element, Mixture, TabulatedProperty
XML Detector Description Editor

- Developed a graphical editor to “hide” XML to the end-users (physicists)
  - It understands our model (DTD)
  - But it’s generic (possible to use another DTD)
  - It understands “links” and allow us to edit a web of XML files as a single tree
  - It’s implemented in Java (portable)
Conditions DB

- Accessing detector conditions data (calibration, slow control, alignment, etc.) should be the same as geometry data
  - Time validity period and Versioning in addition
  - Conditions are integrated into the transient data model
  - Converters are responsible for converting from database rather than from XML
- Conditions are attached to Detector Elements as for geometry
  - each element has many conditions (calibration, alignment, slowcontrol, fastcontrol ...)
Conditions DB Implementation

- Condition objects are stored in XML
- The XML fragments are stored using the Oracle condition DB developed by IT
- XML references are used to select between XML and condition DB:
  - `<conditionref href="/Ecal/condition.xml#caEcal"/>
    → XML`
  - `<conditionref href="/cond://dd/Calibration/Ecal/caEcal"/>
    → DataBase`
Geometry Visualization

- Visualization is essential for developing the geometry
  - Applicable at the different data representations
- Generic geometry information conversion to 3D graphics data
- Panoramix (OnX)
Interfacing With Geant4

- We integrate Gaudi with Geant4 by providing a number of “Gaudi Services” (GiGa)
- The GiGaGeomCnvSvc is able to convert transient objects (DetElem, LVolume, Surfaces, etc.) into G4 geometry objects
  - The conversion does not require “user” code
  - Flexibility in mapping Gaudi model to Geant4 model
- Single source of Geometry information
Status of LHCb Detector Description

- The DetDesc framework is fully functional (transient classes, XML DTD, XML converters, editor, etc)
- All sub-detectors are already described (structure, geometry and materials) using the provided framework
- Visualization based on OnX: Panoramix. Allows to see geometry, events, histograms, ...
- Conversion to Geant4 through GiGa is complete
Example 1: Velo and Rich1
Example (2) : Zoom on Ecal