Detector Description in GAUDI

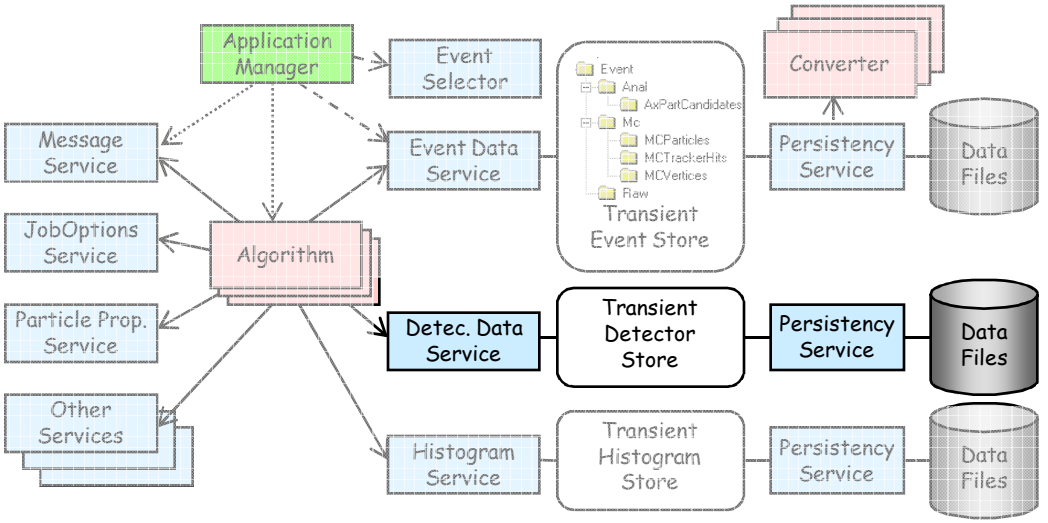
- Architecture
- Detector Logical Structure
- Extending Detector Element
- Summary

Workshop on Geometry Toolkit for the Linear Collider
24th February 2010
P. Mato / CERN

Detector Description Architecture

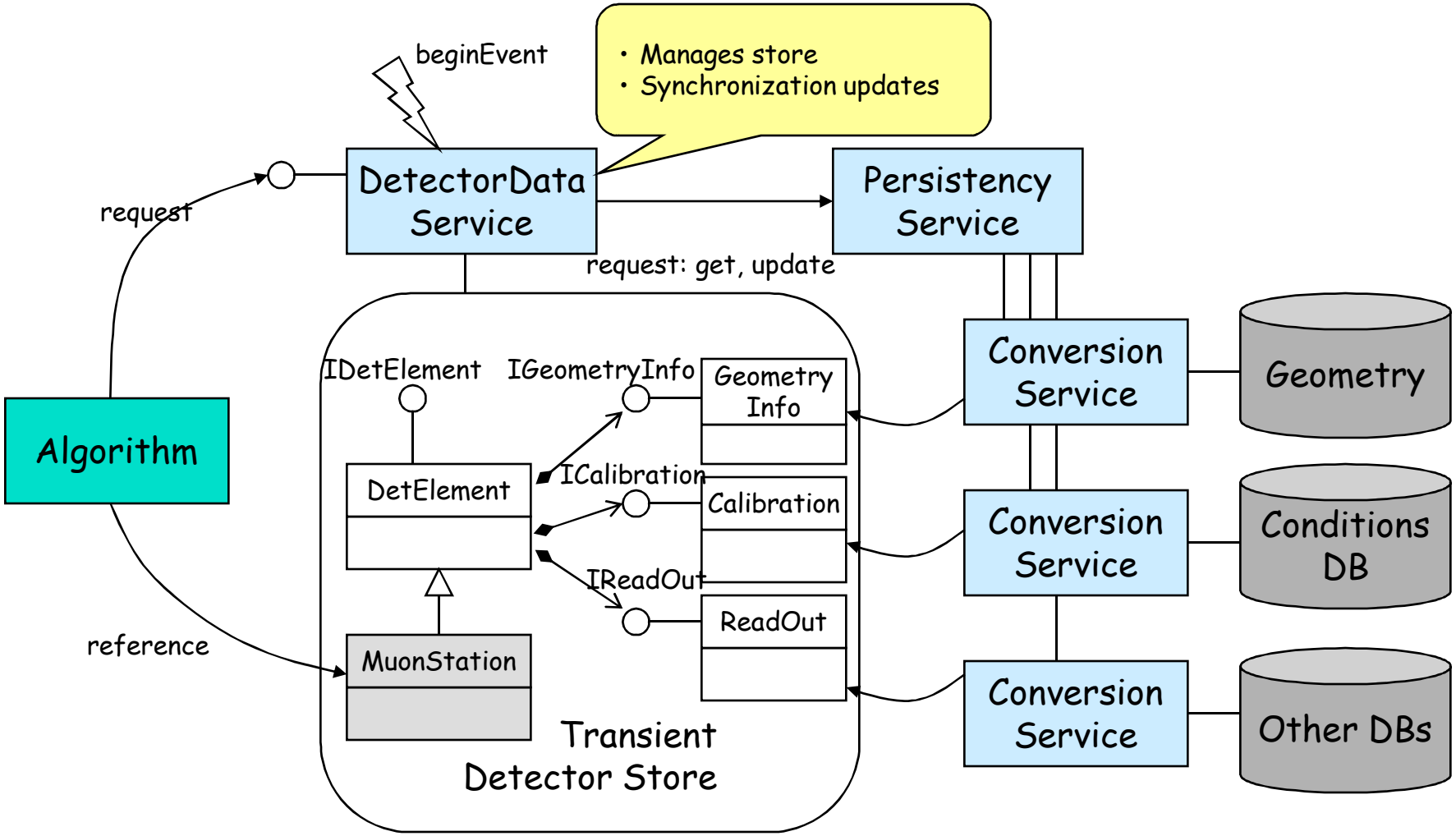
- Sub-Architecture of Gaudi
  - Same principles
  - Transient/Persistent representations
- Focus on the "Physics Algorithm"
  - Access to Detector Transient Store
- Coherent access to "all" detector data
  - Geometry, Calibration, Slow Control, etc.

Gaudi Architecture



CHEP 03 Paper: http://www.slac.stanford.edu/econf/C0303241/proc/papers/THJT007.PDF

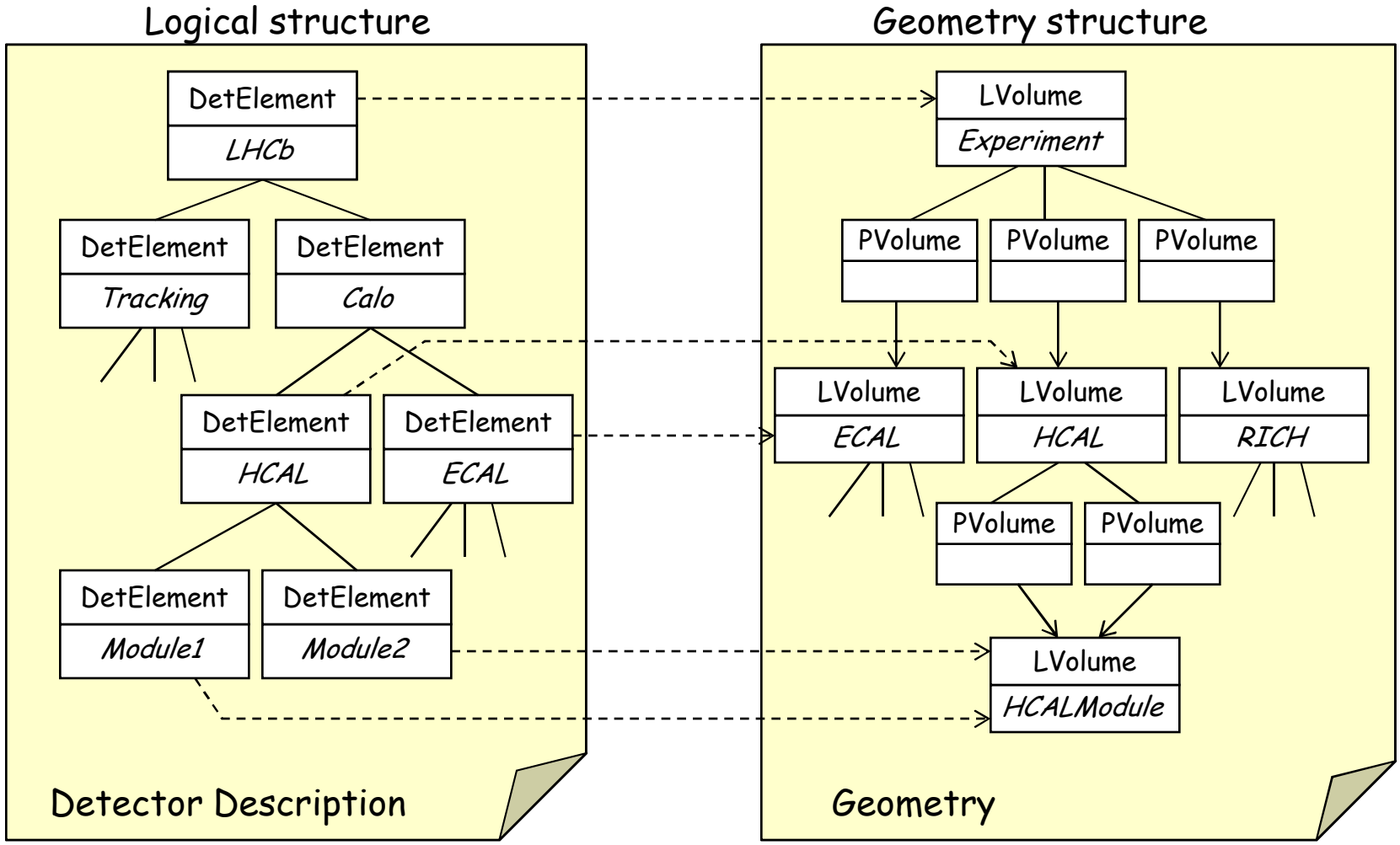
Algorithm Accessing Detector Data



Detector Description

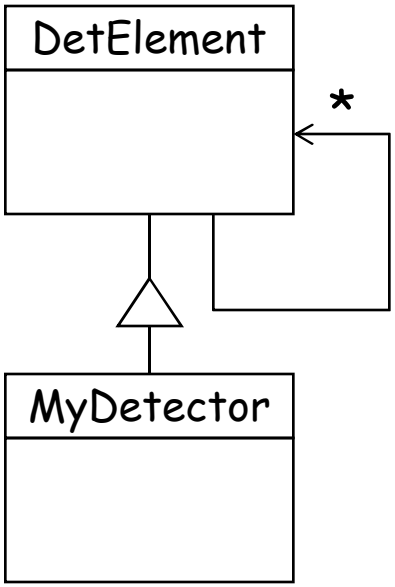
- Logical Structure
  - Breakdown of detectors
  - Identification
- Geometry Structure
  - Hierarchy of geometrical volumes
  - LogicalVolumes (unplaced)
  - PhysicalVolumes (placed)
- Other detector data
  - Calibration, Alignment, Readout maps, Slow control, etc.

Two Hierarchies

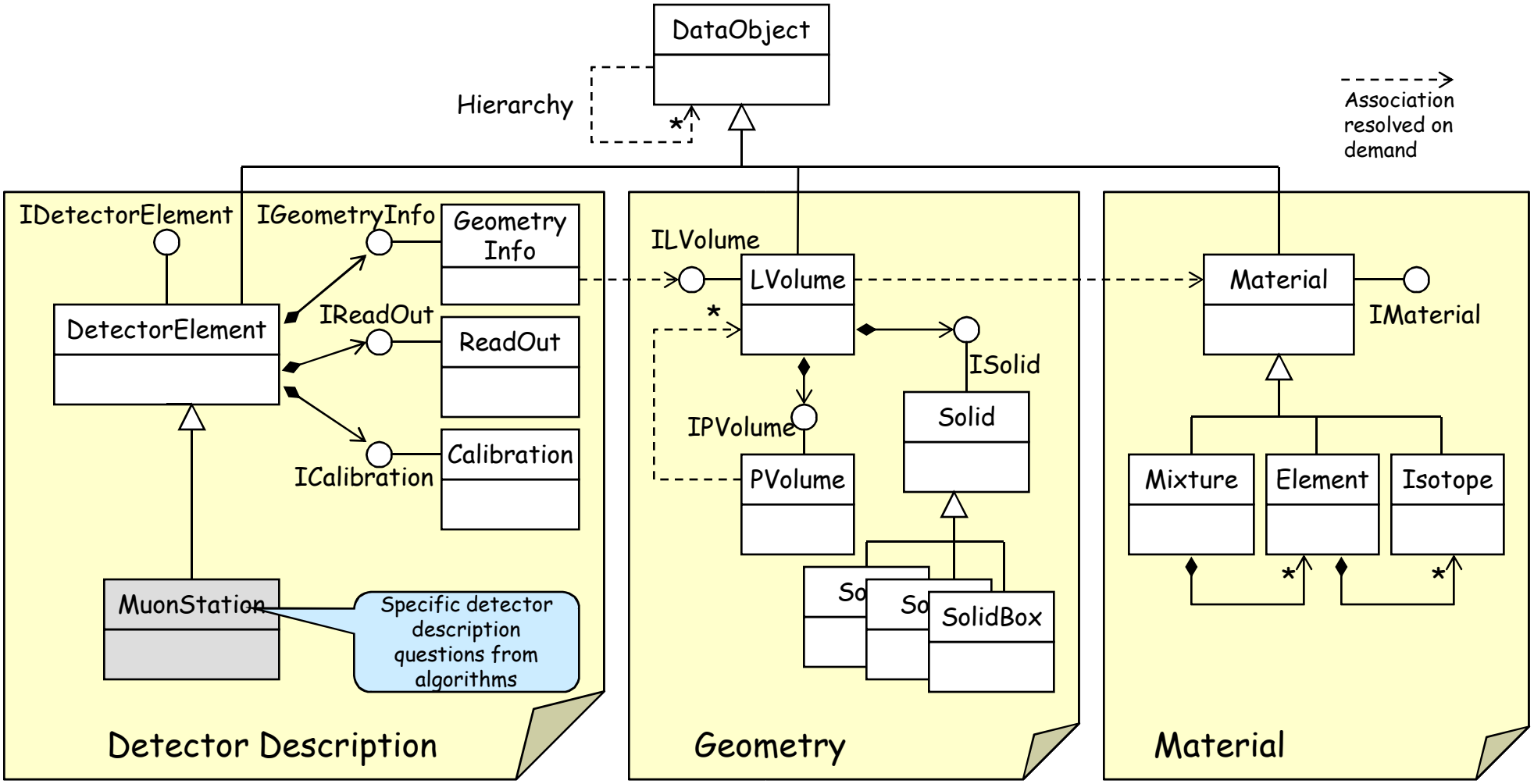


Logical Structure

- The basic object is a *Detector Element*
  - Identification
  - Navigation (tree-like)
- DetectorElement 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 can be coded by "Physicists"
- DetectorElement objects are shared by all Algorithms

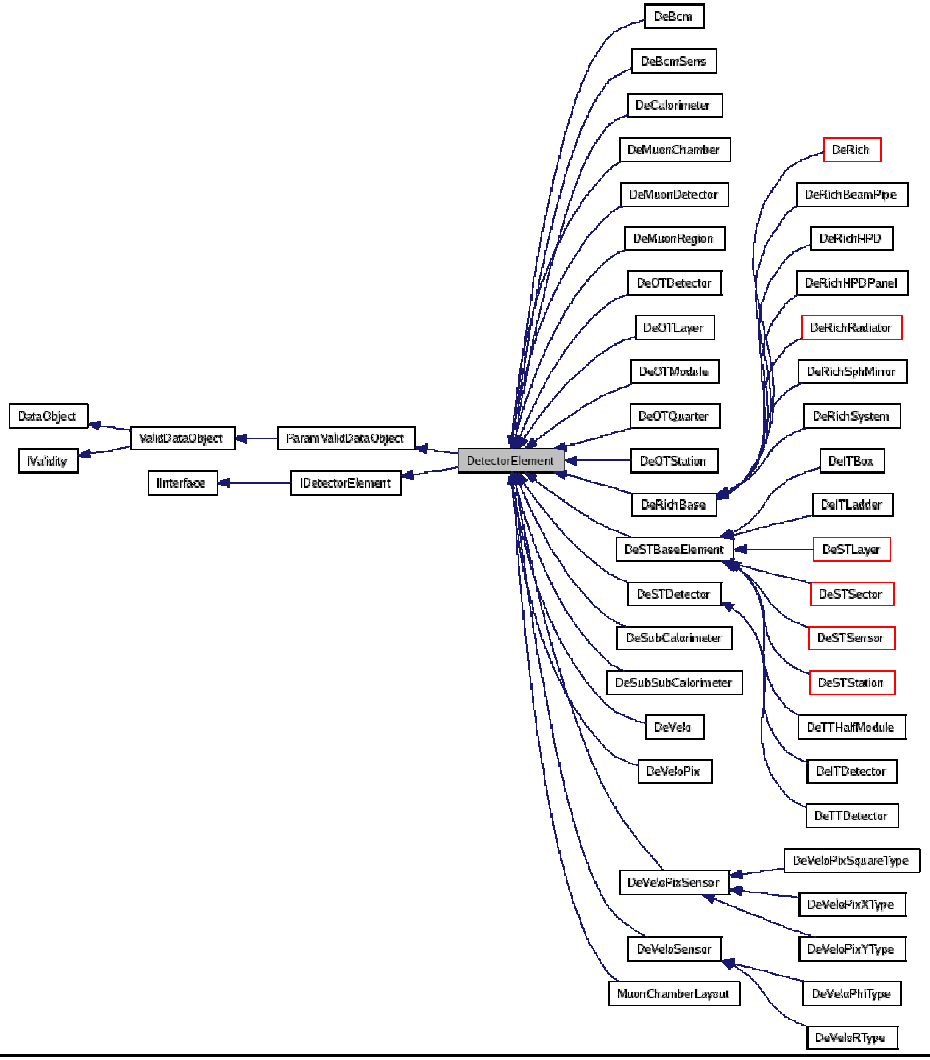


Simplified Diagram (simplified)



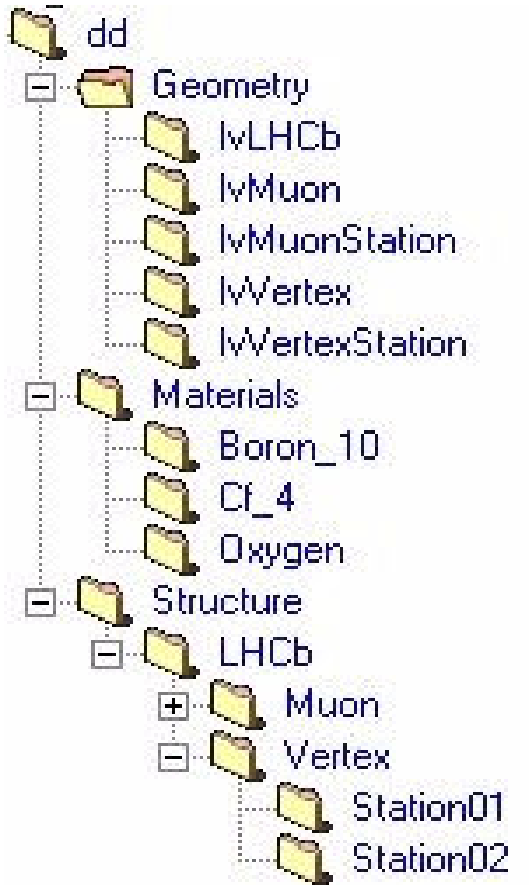
Detector Element Class

- Three basic functionalities:
  - IDetectorElement: Access to other Detector information
  - IValidity: Time validity interval management
  - ParamList: User parameters (key-value pairs)



Transient Store Organization

- Standard Gaudi Transient Store
  - "Catalogs" of Logical Volumes and Materials
  - "Structure" as a tree
  - All elements identified with names of the form: /xxx/yyy/zzzz



Persistency Based on XML Files

- XML is used as persistent representation of the Structure, Geometry and Materials
- Why XML?
  - Instead of inventing our own format use a standard one (extendible)
  - Many available Parsers and Tools
  - Strategic technology

```
<DDDB>
 <catalog name="…">
  <detelem name="…">
   <geometryinfo
        lvname="…"
        npath="…"
        support="…"/>
   <userParameter
        comment="…"
        name="…"
        type="string">
    …
   </userParameter>
   <specific>
    …
   </specific>
  </detelem>
 </catalog>
</DDDB>
```

Specializing Detector Elements

1. Adding userParameter(vector)s to default DetectorElements
2. Extending and specializing the DetectorElement class in C++, using userParameters in XML
3. Extending XML DTD and writing a dedicated converter

Summary

- Detector Element is the central point for offering Detector information to *Algorithms*
  - Can be customized to answer specific questions
    - » Global sub-detector questions should be asked to the Detector Element that represents a sub-detector
    - » Module specific questions should be asked to Detector element that represents a module
- Access similar to any GAUDI Data Transient Store
- Persistency representation based on XML
- Three possibilities for specializing Detector Elements
  - Adding userParameter to default DetectorElements
  - Extending and specializing the DetectorElement class in C++, using userParameters in XML
  - Extending XML DTD and writing a dedicated converter