Handling Several Geometry Configurations in XML

Sebastien Ponce
Friday, 8 March 2002
Overview

- Current Situation
- Handling of several LHCb setups
- How to Handle smaller geometry changes in the future
- A word on parameters and their use
Current Situation

- **Only one geometry** per version of XmlIDDDDB
- One application may only use one version of XmlIDDDDB and thus only one geometry
- No versioning at the level of volumes or even subdetectors
- **No notion of "setups"** at all (LHCb-classic, LHCb-lite, LHCb-minus, ...)
Proposal

- Handle of several LHCb setups inside a single version of XmlDDDB
- Possibility of changing the setup used in an application through a jobOption (DetectorDataSvc.DetDbLocation)
- No versionning at the level of subdetectors or volumes
- No (simple) possibility of using several geometries in the same execution of an application
What Will Change

- Definition of **many entry points** in the database through several files:
  
  \[
  \text{dddb.xml} \rightarrow \text{lhcb.xml, lhcb_lite.xml, ...}
  \]

- Replication of the higher part of the XML file hierarchy

- **Sharing of the subdetector descriptions** between different setups (as a default implementation)

- When subdetectors will provide different implementations for different setups, only few pointers need to be changed
Always Better With a Picture
Smaller Changes

- They will be handled by the condition database.
- This will allow versioning and tagging of every volume in the description.

- Available right now BUT using objectivity.
- ORACLE version being developed by IT, tested last week. Time scale of 6 months to be in production.

- `<catalogref href="conddb:/Geometry/LHCb#LHCb"/>`
Scope of Parameters

- Sharing XML code between different setups includes the sharing of parameters
- Rule: once defined, a parameter is visible in all files down the tree
- Thus: always take care to define it high enough in the tree
Some Rules

- Always put parameter at the **lowest possible level**
- Bring them up if they are used in several files
- Never put parameters out of your subdetector

- Problem when a parameter should be shared in both the structure and the geometry

- Solution: using XML entities
Using Entities

- Idea: you can include files using entities in XML
- One can define parameters in a separate file and include it in both structure and geometry part

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DDDB SYSTEM "../../DTD/geometry.dtd" [ <!ENTITY params SYSTEM "paramfile.xml"> ] >
<DDDB>
  &params;
</DDDB>
```

`paramfile.xml`

```xml
<parameter name="epsilon" value="0.01" />
<parameter name="SiThick" value="0.300*mm" />
<parameter name="RPhiDist" value="1.0*mm+SiThick" />
```