LHCb Detector Description

DDDB Meeting
23 October 1998
P. Mato, CERN
Foreword

We have started recently (4 weeks ago) to design the architecture of the LHCb data processing applications.

- Simulation, reconstruction, analysis, high level triggers, etc.
- The ideas are still very preliminary.
- The detector description not in the critical path.

We will try the ideas before end of the year.

Operational detector description needed by spring 1999.
Basic architectural choices

- Clear separation between “data” and “algorithms”
- Three basic types of data: event data, detector data, statistical data.
- Clear separation between “persistent data” and “transient data”. Data centered architectural style.
- *User code* encapsulated in few specific places: “Algorithms” and “Converters”
- All components with well defined “interfaces” and as “generic” as possible.
This week Architecture

Diagram showing the architecture of a system with various components and their relationships. The components include:

- **PersistencySvc**
- **MessageSvc**
- **JobOptionsSvc**
- **AppManager**
- **AlgFactory**
- **EventSelector**
- **EventDataSvc**
- **ObjContainer**
- **Obj1**, **Obj2**, **Obj3**
- **Alg1**
- **Alg2**
- **Alg3**

The diagram also shows the transient event store and the detector store, with arrows indicating the flow of data and interactions between components.
## Classification of Classes

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Managers</td>
<td>One per application. The chief of orchestra.</td>
</tr>
<tr>
<td>Services</td>
<td>Offering specific services with well-defined interfaces. Different concrete implementations depending with specific functionality.</td>
</tr>
<tr>
<td>Converters</td>
<td>In charge on converting specific event or detector data into other representation.</td>
</tr>
<tr>
<td>Selectors</td>
<td>Components to process a selection criteria for events, parts of events or detector data.</td>
</tr>
<tr>
<td>Event/Detector data</td>
<td>The data types that the algorithms and converters are using. No complex behaviour.</td>
</tr>
<tr>
<td>Utility classes</td>
<td>All sort of utility classes (math &amp; others) to help on the implementation of the algorithms.</td>
</tr>
</tbody>
</table>
Architecture (class diagrams)

Services

- Service
  - IService
  - IPersistencySvc
  - IQueryOptions
  - IEvtDataSvc
  - IMessageSvc
- PersistencySvc
- JobOptionSvc
- EvtDataSvc
- MessageSvc
- EventPersSvc
- DetPersSvc
Detector Data Store

- Detector Elements need to be “identified”
- Strong hierarchical structure
- Versioning
Detector description: Visualization

- Transient Event/Detector Store
- Rep. Service
- Converter
- Selector
- User Interface
- Mark objects in Store

Representations Store (graphical, textual)
Detector Description

- It includes:
  - Detector structure (final detector, test beam, etc.)
  - Geometry & Positions (Ideal, Real, Simulation). Versioning based on time, run #, etc. Material.
  - Mapping electronic channels to detector cells. Dead channels.
  - Detector control data needed for reconstruction (time based).
  - Calibration and alignment data.

- The transient detector store contains a “snapshot” of the detector data valid for the event currently being process.
Links between Event/Detector

- A priori different “persistent stores”. Logical identification needed.
Summary

- Software architecture currently being designed. Strategical choices being done.
- The different views of the “detector description” will be implemented going through the “transient data store” using a set of converters.
- The Detector Description database will contain a big variety of data related to the detector. Powerful editing tools will be required.
- The schema of the database will be needed soon.