Strategy for Migrating the LHCb Software to the GAUDI Framework

Pere Mato, CERN
7th October 1999
Final objective:
- Produce a complete set of fully functional data processing applications using exclusively OO technology.

Sub-objectives:
- Provide a fully functional framework (GAUDI)
- Assemble new and old algorithms into a single and complete suit of data processing applications. Be able to run productions.
- Convert all the existing FORTRAN code to C++
Possible strategies

1. SICb → Gaudi
2. SICb → Gaudi → Fast translation of Fortran into C++
3. SICb → Gaudi → Wrapping Fortran

Framework development phase, Transition phase, Hybrid phase, Consolidation phase.
Framework development phase

- At the end of this phase the GAUDI framework should be functionally complete
  - Data access services
  - Generic event model
  - Generic detector description model
  - Data visualization
  - Basic set of services

- Develop some physics algorithms to prove architecture concept
- We started this phase one year ago
- We expect to be completed by middle of November
Transition phase

- At the end of this phase we should be able to reconstruct and analyze simulated data within the GAUDI framework. The Monte Carlo data production will still be done using Sicb.
- Incorporate reconstruction and analysis parts of SICb in the GAUDI framework - **wrap FORTRAN code**
  - Analyse SICb to identify all modules, their inputs and outputs
  - **Develop a complete OO event data model**
  - Write converters to allow access to data in both formats
- Development of new algorithms can proceed within GAUDI
- Caveats
  - lot of work to make converters in both directions
  - we could discover technical difficulties (size, commons, initialization,...)
Transition Phase (2)

Diagram showing the flow of data and processes:

- Generator Det. Simul. (Sicb) -> MC Hits
- MC Hits -> Commons
- Commons -> FA, FB, FC
- FA, FB, FC -> Bank 1, Bank 2, Bank 4 (ZEBRA)
- Bank 1, Bank 2, Bank 4 -> Cnv
- Cnv -> Obj 2, Obj 3, Obj 4 (GAUDI store)
- Obj 2, Obj 3, Obj 4 -> C++A, C++B
- C++A, C++B -> Recons + Analysis

Data flows to:

- DST
- Histo Ntuples

Date: 7/10/99
Hybrid Phase

- One single program with FORTRAN and C++ cooperating to produce physics results.
- Replace wrapped FORTRAN code incrementally.
- At the end of this phase we should be able to retire the FORTRAN compiler and libraries.
- Already known problems:
  - Two different detector descriptions. Difficult to maintain.
  - Output file format. Which one to choose?
  - Different set of input “card files”.
  - Huge memory needs for data and code.
- The hybrid phase should be as short as possible to minimize the pain.
Consolidation phase

- Complete the new detector description
- Re-iterate with the O-O Event Model
- Re-engineer some of algorithms
- Incorporate Geant4
- etc.
## Planning

<table>
<thead>
<tr>
<th></th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Qtr 3</td>
<td>Qtr 4</td>
<td>Qtr 1</td>
</tr>
<tr>
<td>Architecture Design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaudi Development v1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaudi Development v2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaudi Development v3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Framework Functional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis Sicb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transition phase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production program</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid phase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Migration completed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>