LHCb Offline Application Framework

Status and planning
29 September 1998
P. Mato, CERN
Driven by Requirements

◆ Functionality release 1.0:
  - Object Oriented environment that allows user to:
    » Define input and output data, job parameters (c.f. SICB.dat)
    » Loop over events
    » For each event, access MonteCarlo truth and digitised raw data
    » Output results in the form of HBOOK histograms and/or ntuples
    » Provide placeholders user initialisation and analysis code (c.f. suinit, suanal)
  - Does NOT allow user to:
    » Store back into ZEBRA store (can be discussed...)
    » Access SICB reconstruction output
    » Use an analysis library (c.f. AXLIB)
  - Input is from ZEBRA files produced by SICB

◆ Typical “client”:
  - Sub-detector expert developing a pattern recognition algorithm. Release includes example of user code
Driven by Qualities

- Discerned by observing the system at runtime
  - Performance
  - Availability (proportion of time the system is up and running)
  - Usability (learnability, efficiency, memorability, error avoidance, error handling, satisfaction)

- Not Discerned by observing the system at runtime
  - Modifiability
  - Portability
  - Reusability
  - Integrability
  - Testability
## Project planning

The architecture affects the structure of the development team.
- It defines the units of software (modules), which are the work assignments.
- The architecture may affect the requirements for the next version.

### Task Planning

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
<th>January</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User Requirements</td>
<td>24/8</td>
<td>31/8</td>
<td>7/9</td>
<td>14/9</td>
<td>21/9</td>
</tr>
<tr>
<td>2</td>
<td>Architecture Design</td>
<td>28/9</td>
<td>5/10</td>
<td>12/10</td>
<td>19/10</td>
<td>26/10</td>
</tr>
<tr>
<td>3</td>
<td>SDE preparation</td>
<td>2/11</td>
<td>9/11</td>
<td>16/11</td>
<td>23/11</td>
<td>30/11</td>
</tr>
<tr>
<td>4</td>
<td>Module Development</td>
<td>7/12</td>
<td>14/12</td>
<td>21/12</td>
<td>28/12</td>
<td>4/1</td>
</tr>
<tr>
<td>5</td>
<td>Module Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Coding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Module testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>System Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Release 1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

29/9/98
First step: Design Architecture

- Definition:
  - The software architecture of a computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.

- We will define 3 architecture views (or structures)
  - Module structure (Units are work assignments)
  - Uses structure (Units are programs or modules with "using" relation)
  - Data flow structure (Units are programs or modules with the relation may-send-data-to)

- Architecture team

- Regular architecture reviews
Class Category decomposition
Preliminary Ideas of the Architecture
Identified Components

- Application Manager
- Job Options Service
- Event Persistency Service
- Detector Persistency Service
- Message Service
- Detector Persistency Service
- Graphical Representation Service
- Algorithm interface
- Transient Event data model
- Transient Detector data model
Software Development Environment

- Development platform: NT
- Design tool: Rational Rose
- Coding/debugging: Visual C++
- Code Management: Visual SourceSafe
- Code repository: `\alnts1\Packages\LHCb`
- Documentation: ?
- Web authoring: Front Page 98
Conclusions

- We just started last week
- We are convinced of the importance of the architecture
  - Communication among end users, managers, development team
  - Early design decisions
  - Most of the qualities can be evaluated with the architecture
- We will inform of progress at weekly basis. Feedback is needed.
- We will Just Do It