Accelerator Data Foundation: How It All Fits Together

Ronny Billen

Pascal Le Roux, Maciej Peryt, Chris Roderick, Zory Zaharieva

CERN, Beams Department
Controls Group
Data Management Section

The 12th International Conference on Accelerators and Large Experimental Physics Controls Systems
October 12-16, 2009
Kobe International Conference Center
Kobe, Japan
Outline

- Introduction
- A Vision, a Strategy and some Tactics
- Accelerator Data Management Domains
- Data Federation
- Covering the CERN Accelerator Complex
- Human Resources
- Conclusions
Introduction

Two major events can be considered as turning points in data management in CERN’s Accelerator Sector:

1. Purchase of a commercial RDBMS in 1983
   - Aimed to address complex technical aspects of LEP
     - Project planning, cabling, documents, magnet data,…
   - Since then, many successful database driven systems were implemented, but in a dispersed way
     - PS Controls, LEP Alarms,…

2. Reunification of “Accelerators & Beams” activities in 2003
   - Single groups for Operation, Controls and Equipment for the complete CERN accelerator complex
   - Unification of accelerator data management was proposed…and accepted by the hierarchy
The Vision

Data Management Team in the Controls Group

- Dedicated team of 8-10 database software engineers
- Senior members having also accelerator domain knowledge
- Junior members proficiently up-to-date in database development
- Ambitious technical and human objectives for the team

Need for a clear vision

- Highest importance attributed to LHC
  - huge complexity, but starting from a relatively clean sheet
- Rationalize, improve and federate the existing data on the older accelerators
- Break up the vast domain into manageable areas
  - Typically a senior and junior team member in each area

Communicate, communicate, communicate
The Puzzle

Breaking up the data domain into several pieces

- Advantages
  - Logical break-down
  - Organized
  - Manageable

- Inconveniences
  - Integration not considered at the outset

- Therefore
  - Emphasize common understanding and good communication between developers
  - Federation work began in 2007
Strategy

✦ Development work
   ☐ Major effort: requirements gathering, analysis and development work
   ☐ Legacy DB showed lack of *Quality Assurance* at DB and data levels
      ☐ Introduced integrity constraints and naming conventions

✦ Technology
   ☐ Use of Oracle technology stack (huge in-house expertise)
   ☐ Use of Java, J2EE, JDBC deployed in 3-tier architecture
   ☐ On-line usage of database services for accelerator control

✦ Responsibilities
   ☐ Acceptance of competence shift with clear limits of responsibilities
     ① DM team developers
     ② Application developers
     ③ Data owners
     ④ DB infrastructure service
Some Tactics

- The important aspects to make the strategy successful
  - Involve end-users right from the start, throughout the design and development process
  - Communicate constantly on scheduled interventions and their anticipated impact
  - Iterate rapidly based on end-user feedback
  - Provide adequate environments for development, unit testing, system testing and production
  - Push data ownership to the experts, assist and guide the usage of the data maintenance interfaces
Layout Data

1. **Machine Layout**
   - Accelerator design, magnetic model, beam optics by machine physicists
   - Mechanical installation & integration work; establishing the as-built model

2. **Controls Electronics Layout**
   - Racks, crates, modules, fieldbus connections
   - 9,000+ racks for LHC
   - Essential during installation
   - Starting point for automatic \textit{configuration} of front-end computers

3. **Electrical Circuits Layout**
   - Description of electrical objects in powering circuits
   - Relationship between power converters, current leads, bus bars, magnets, ...
   - Exploited in \textit{operational data} domain

\textit{Icalepcs'07 RPPA03}
**Controls Configuration**

- **Control system topology**
  - From front-end computers to control room consoles
  - 65,000+ controls devices
  - 5 device-property models
  - Rejuvenated legacy DB

- **Interactive interfaces and APIs**
  - 200+ re-developed Oracle ADF interfaces (Java-based forms)
  - 150+ Oracle APEX reports
  - Java APIs for control room apps
  - Pro*C APIs are being replaced
Operational Data

1. **Settings**
   - Parameter space for LSA

2. **Measurements & Logging**
   - Store time-series data, keep on-line
   - Beam and equipment measurements

3. **Alarms - LASER**
   - Capture, store, notify anomalies
Physical components

- CERN-centralized asset management
- CERN-wide part identifier
- Commercial system enriched with home-made interfaces
- Keep track of complete lifecycle of the asset
Fitting the pieces of the puzzle

- QA starting point: identification of objects
  - Equipment code catalogues as official references

- Supporting QA tools
  - Including a Naming database

Basis for Data Federation

- Imposed naming conventions
- Clear procedures, standards and instructions to be followed by all
The perfect solution for data integration…

- **Unique identifier** (i.e. primary key) for each object throughout all data domains

…was not put in place, so…

- How do we solve this problem?

Exchange of keys between the domains

- Object Identifiers are truly **unchangeable** primary keys
  - Not the object name! It may change over time

- Several implementations on a case-by-case basis
  - Database views, materialized views
  - Use of ‘grant select’ or database links
  - PL/SQL code for more complex data propagation
  - Execution on manual, semi-automatic or automatic basis

Best illustrated by example
Asset identifiers are imported

Asset-to-Slot assignment done by the equipment owner via this interface

Slot identifiers are imported
Other Examples of Data Propagation

- Beam loss monitoring electronics
  - Module data needed for beam interlocks
- Commissioning sets of circuits
  - Electrical layout needed to drive settings
- Configuring front-end computers
  - HW/SW configuration derived from layout data
- Driving settings from SW devices
  - From device-property to Device-parameter
- Generating alarm definitions
  - Based on controls configuration
Architecture, design and implementation was set out with LHC in mind

Retrofit to existing accelerators is in progress
  ➔ Integrate the existing data into the current model
  ➔ Extend the model to cater for specifics of other accelerators
  ➔ High level controls and settings management of the PS-complex is the most difficult challenge
  ➔ Renovation project with convergence towards LHC has been launched
The people are the most important assets in the process of analysis, design, development and maintenance. The core team of database engineers have to:

- Follow and use the technology effectively
- Acquire specific domain knowledge
- Show flexibility in adapting to the changing user requirements
- Impose data access methods to application developers

Their responsibility is proportional to the database complexity.

<table>
<thead>
<tr>
<th></th>
<th>tables</th>
<th>constraints</th>
<th>code</th>
<th>volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
<td>134</td>
<td>495</td>
<td>55,602</td>
<td>5.3 GB</td>
</tr>
<tr>
<td>Configuration</td>
<td>514</td>
<td>1,524</td>
<td>30,326</td>
<td>9.7 GB</td>
</tr>
<tr>
<td>Settings</td>
<td>281</td>
<td>1,392</td>
<td>9,026</td>
<td>14.6 GB</td>
</tr>
<tr>
<td>Logging</td>
<td>55</td>
<td>103</td>
<td>14,431</td>
<td>+17 TB</td>
</tr>
<tr>
<td>Alarms</td>
<td>207</td>
<td>191</td>
<td>24,915</td>
<td>62.8 GB</td>
</tr>
</tbody>
</table>
What’s the important message here?

- Data management is an **organizational** issue
- The accelerator domain is a **very wide area**
  - Legacy, in-house developments, commercial systems are part of our environment
- Ensure **single source** of maintained, consistent data
  - Avoid confusion, doubt and errors
- In case of data propagation
  - Procedures for execution and verification of the data synchronization are mandatory (i.e. Quality Assurance)

- A database is only as good as the correctness of the data it contains
- Software comes and goes, data stays forever
Questions?