An Integrated Control System for the LHCb experiment

Alba Sambade Varela
On behalf of the LHCb online group

16th IEEE NPSS Real Time Conference 2009
May 10-15 IHEP Beijing
Outline

- LHCb
- Starting point
- ECS architecture
  - Finite State Machines
- Implementation
  - Subsystems
  - Run Control
- Conclusions
LHCb experiment

- It is one of the 4 detectors in the *Large Hadron Collider*
- Its aim is studying the production of pairs of *b* quarks (*beauty*-antibeauty) and the CP violation (asymmetry matter/antimatter).
- It is composed out of 5 groups of subdetectors:
  - VELO
  - RICH
  - Tracker
  - Calorimeters
  - Muon system.
- Each one focused in one property.
The ECS handles the configuration, operation and monitoring of every online (run time related) task in the experiment.

It has been implemented as a hierarchical control, which is

- Homogeneous

and provides:

- Automation
- Partitioning

LHCb Experiment Control System

Data flow

Controls flow
Common tools: Homogeneity

- Starting from a commercial SCADA software: PVSS,
- a Framework Toolkit,
  - Set of guidelines and sw tools, developed in common for the four LHC experiments.
- and a FSM package,
  - Allow the creation of hierarchies of Finite State Machine.
- a homogeneous design was achieved for the ECS.
  - The same behavior schema is used through the system to model all control nodes.
  - All data is stored in a run time database, allowing different external applications to interface it.
  - Common look & feel.
- Single system principles extrapolated to all systems!

The same behavior schema is used through the system to model all control nodes.
- All data is stored in a run time database, allowing different external applications to interface it.
- Common look & feel.
- Single system principles extrapolated to all systems!
ECS architecture

- Hierarchical (tree like) structure
  - A structure of logical nodes integrates every item to be controlled (hardware device, software task, logical entity).
**FSM concept**

- FSM allows the definition and operation of hierarchies of objects (control nodes) behaving as Finite State Machines.
  - It is based on SMI++.

- The behavior of the nodes and its interactions are modeled in terms of STATES and COMMANDs.
  - Change process between states are TRANSITIONS.
  - These changes are induce either by sending a COMMAND or by a state change of a children.
Control nodes types

- **Control Unit**
  - Implements the specific behavior and takes local decisions (automation and sequencing of actions, error recovering)
    - “Expand” actions.
    - Partitioning: included/excluded.
    - Can run in stand-alone mode.

- **Device Unit**
  - Interface with the real devices (hardware or software)
    - Calculates a STATE from device readings.
    - Implements COMMANDs as device parameters.
    - Enabled/disabled.
IT_ECS_St1A: dist_1:Manager2

System | State
-------|-------
IT_ECS_St1A | NOT_READY

Sub-System | State
-----------|-------
IT_DCS_St1A | NOT_READY
IT_DAO_St1A | NOT READY
IT_HV_St1A | NOT READY

Messages

Close
Partitioning (1)

- Different sub-detectors, many teams and possible configurations (operation modes) (commissioning stage).
  - Possibility to run hierarchies in parallel.
  - Possibility to modify dynamically the components included to control.
    - Typical case for an out of order device.
  - Different partitioning modes (included, excluded, manual and ignored).
Partitioning (2)

- **Taking or releasing control**
  - The operator can take the control at an intermediate level of the control tree.
  - Ownership: it is guaranteed that only one user at a time can use a given part of the system.
    - Exception: shared mode, operator with certain rights can also send commands.

- Once a sub-set of the system is excluded, it can be taken separately and run in parallel (stand-alone) with other tree(s).
Automation

- Need coming from the complexity of the system and the operation by non-expert operators.
  - Sequencing of actions
    - An action on control units is specified by a sequence of instructions, mainly consisting on commands sent to their children and logical tests (on the states of these).
    - An action on device units is typically sent off as a message to the real hardware (which can cause a status change and will trigger the corresponding logical checks).
  - Error recovery
    - Logical state checks can trigger automatic actions whenever a malfunctioning is detected.
ECS implementation

- Sub-system integration
  - Sub-detectors
  - Common resources

- Set of guidelines proposed to unified procedures
  - Split sub-detector ECS into 4 identified control domains
Sub-detector ECS

- Control domains
  - They cover all the activities necessary to manage, to supervise and to run a sub detector.
    - DAQ: all Electronics and components necessary to take data (run related).
    - DCS: infrastructure (Cooling, Gas, Temperatures, pressures, etc) that is normally stable throughout a running period.
    - HV: equipment that normally depends on the LHC machine state (fill related).
    - DAI: Infrastructure necessary for the DAQ to work (computers, networks, electrical power, etc.) in general also stable throughout a running period.
  - Every domain has an specified FSM type with standardized states and actions.

- Further domain tree breakdown
  - Physical or logical (up to the subdetector).
Sub-systems integration
Parallel run controls

- Partitioning
  - Alternate control trees in the same control system.
Dynamic allocation

- ReadOut Partitioning
- Reserve a “slice” of common Resources
Run Control

- **Matrix Domain**
- **Sub-Detector**

- **Activity**
  - Used for configuring all Sub-Systems
Conclusions

• We have implemented the LHCb ECS:
  – Highly distributed
    • Control spread over 150 PCs (Windows and Linux).
      – ~2000 CU and ~30000 DU
  – Hierarchical
    • All “equipment” integrated

• And a daily used Run Control
  – Minimal operator intervention (~2 persons shift crew)
    • Automation: actions sequencing and error recovering
  – Non-expert user friendly
  – Subdetectors can run different instances in parallel.
Questions?