ACEOLE
www.cern.ch/aceole

CMS – The Detector Control System

Yi Ling Hwong

ACEOLE Six Month Meeting

This research project is supported by the Marie Curie Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE)

1 April 2009

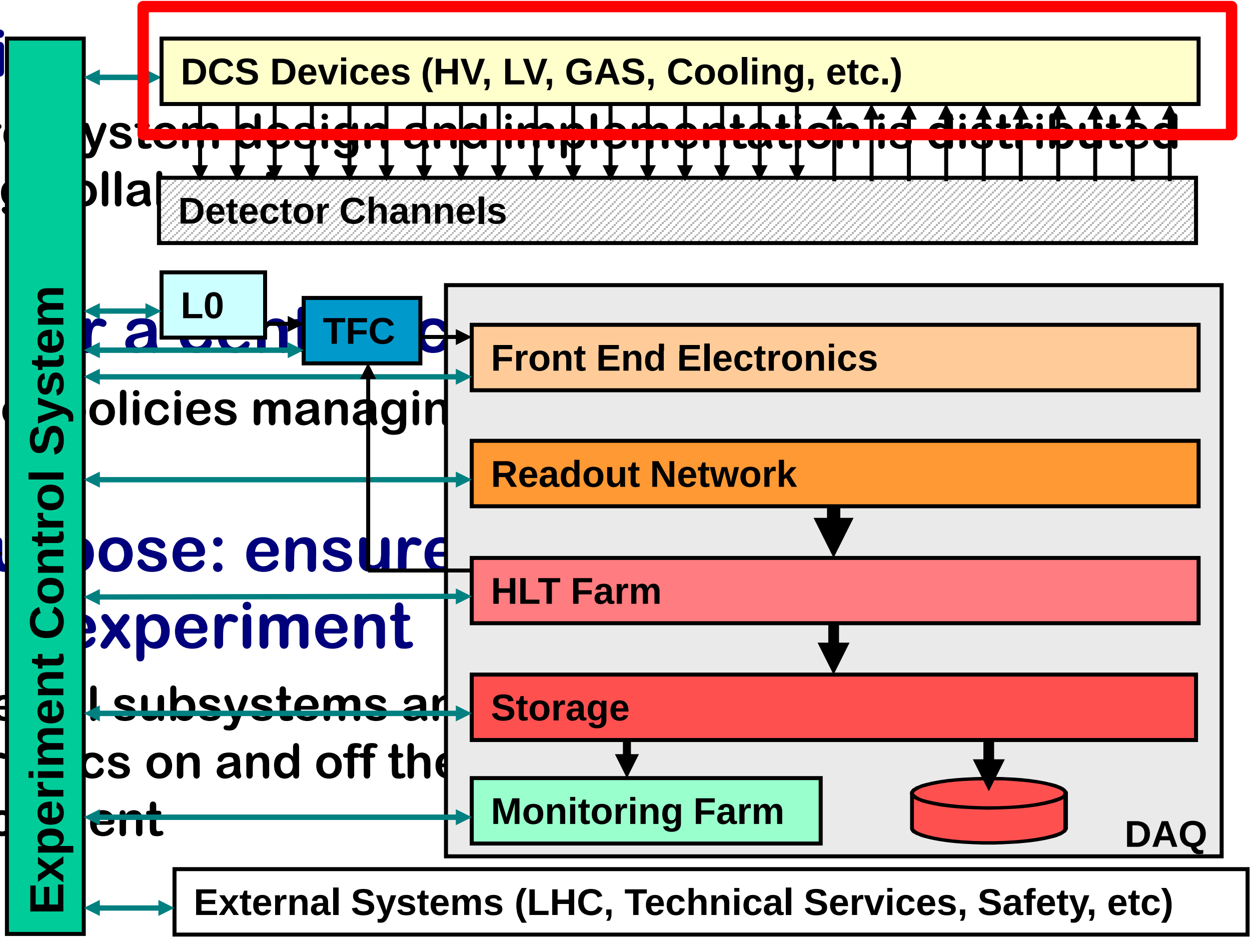
Overview

- Introduction to the CMS Detector Control System
- PVSS and JCOP Framework
  - □ Why and how?
- Finite State Machine
  - □ Tools and Applications
- Analysis of current FSM tree
  - □ Background and objectives

The Detector Control System (DCS)

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

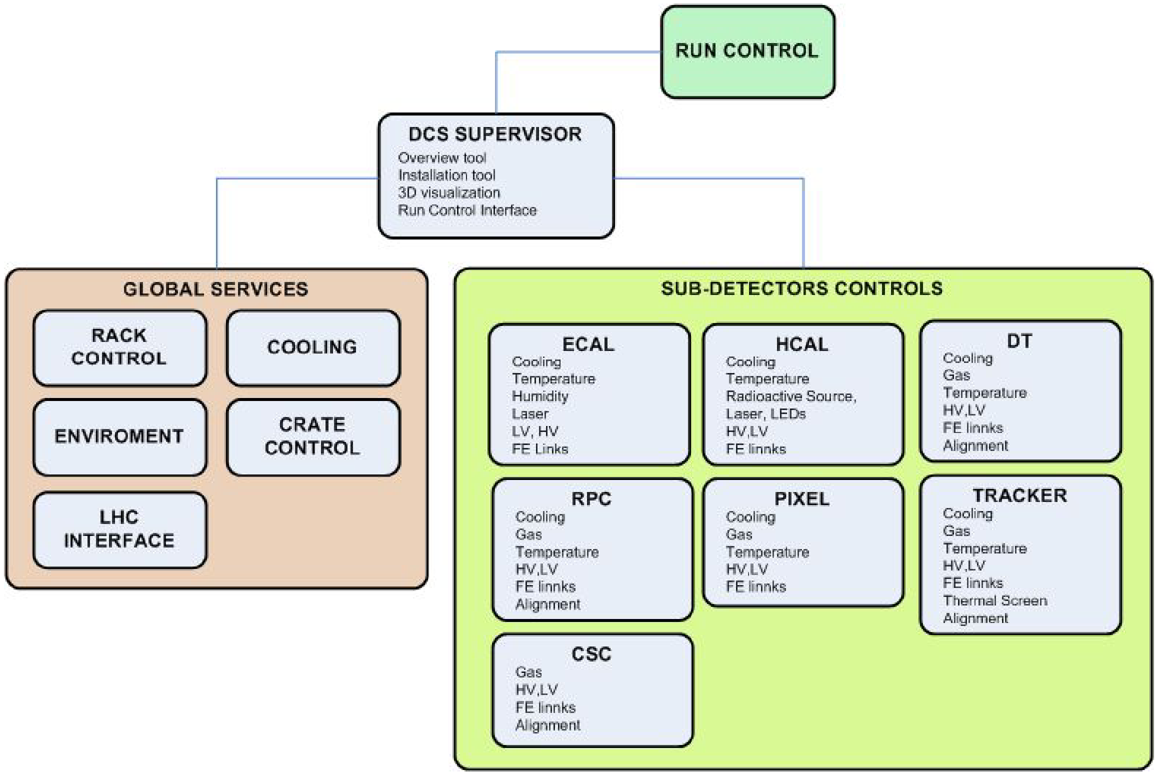
- CMS is a collaboration of more than 35 countri
  - □ Contr ystem design and implementation is distributed among olla
- A need r a
  - A t olicies managin
- Main pu pose: ensure the CM xperiment
  - □ Scope l subsystems ar electr cs on and off the enviro ent


DCS Devices (HV, LV, GAS, Cooling, etc.)
Detector Channels
Experiment Control System
L0
TFC
Front End Electronics
Readout Network
HLT Farm
Storage
Monitoring Farm
DAQ
External Systems (LHC, Technical Services, Safety, etc)

The CMS DCS

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

- The DCS of individual sub-detectors are connected to the central DCS Supervisor
  - □ Global Services
  - □ 7 Sub-Detectors Controls



RUN CONTROL

DCS SUPERVISOR
Overview tool
Installation tool
3D visualization
Run Control Interface

GLOBAL SERVICES

RACK CONTROL

COOLING

ENVIROMENT

CRATE CONTROL

LHC INTERFACE

SUB-DETECTORS CONTROLS

ECAL
Cooling
Temperature
Humidity
Laser
LV, HV
FE Links

HCAL
Cooling
Temperature
Radioactive Source,
Laser, LEDs
HV,LV
FE linnks

DT
Cooling
Gas
Temperature
HV,LV
FE linnks
Alignment

RPC
Cooling
Gas
Temperature
HV,LV
FE linnks
Alignment

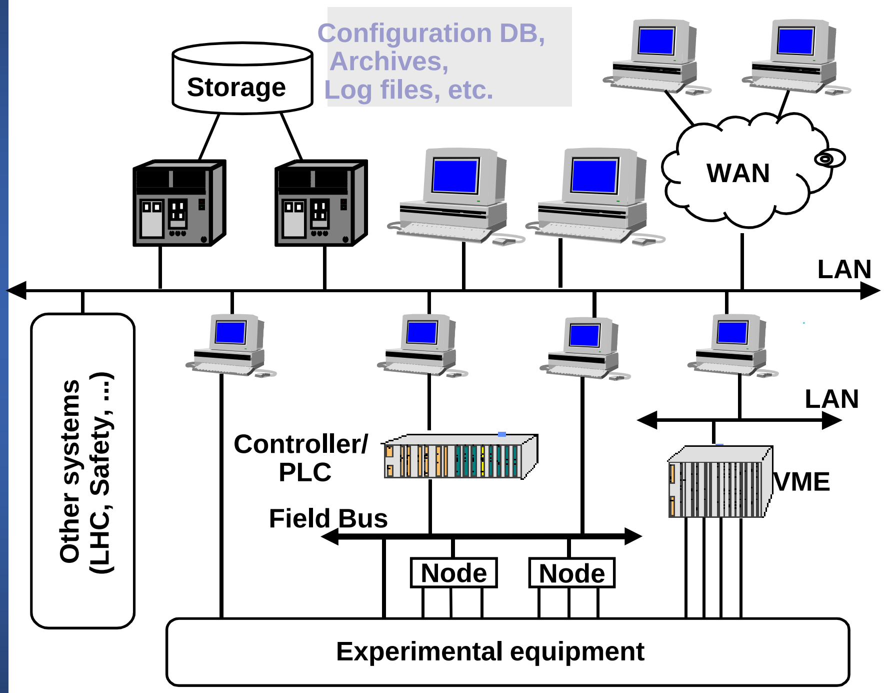
PIXEL
Cooling
Gas
Temperature
HV,LV
FE linnks

TRACKER
Cooling
Gas
Temperature
HV,LV
FE linnks
Thermal Screen
Alignment

CSC
Gas
HV,LV
FE linnks
Alignment

Control Technologies from LHC Era

- **DCS**
- **PVSS/JCOP Framework**
- **Finite State Machine**
- **DEN**
- **Background and Objective**
- **Conclusion**



Layer Structure

- Supervision
- Process Management
- Field Management

Technologies

Commercial | Custom

- FSM
- SCADA (PVSS II)
- OPC | DIM
- Communication Protocols
- PLC/UNICOS | VME
- Field buses & Nodes
- Sensors/devices

Based on an original idea from LHCb

Requirements

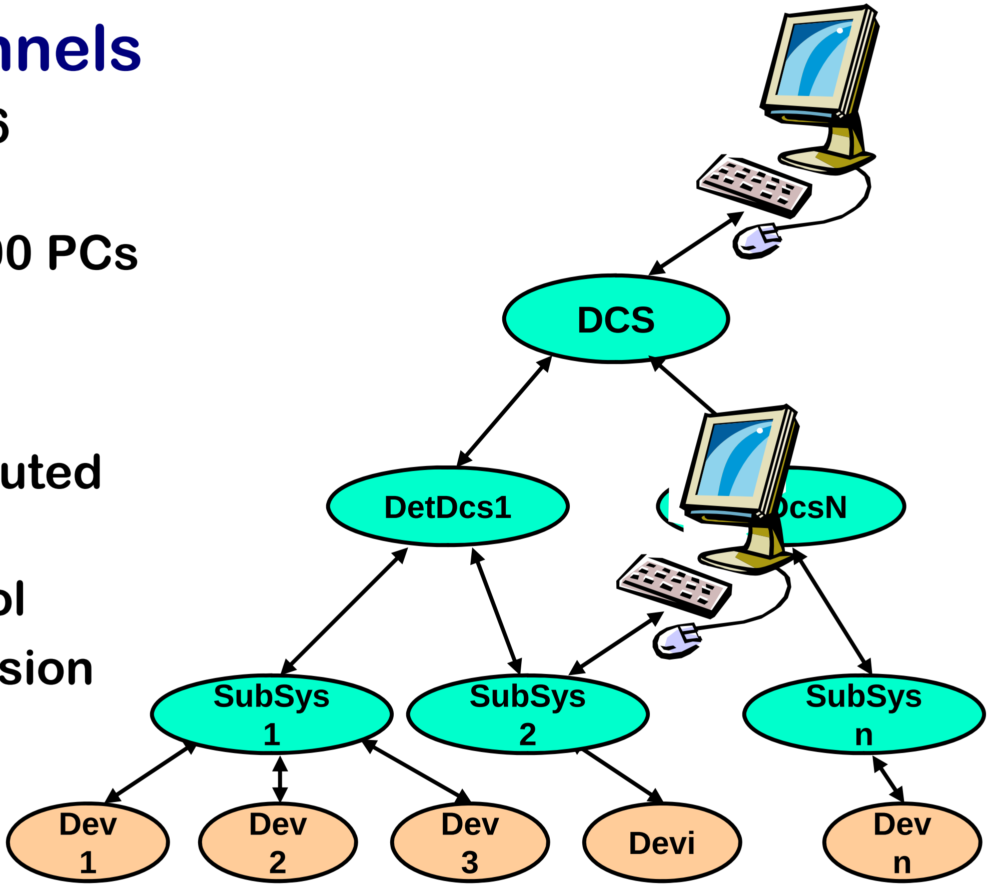
- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

■ **Large number of devices / IO Channels**

- □ For CMS - Around 6 million parameters
- □ Distributed over 100 PCs

■ **Need for**

- □ Parallel and distributed control system
- □ Hierarchical control
- □ Decentralized decision making



What is JCOP?

- DCS
- **PVSS/JCOP Framework**
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

- Stands for “Joint COntrols Project”
- Grouping of representatives from the 4 big LHC experiments
  - □ Offer common experiment components and support (e.g. CAEN, ELMB etc)
- Aims to reduce the overall manpower cost required to produce and run the experiment control systems
  - □ Reduce knowledge of tools
  - □ Interface for non-experts

What is JCOP Framework?

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

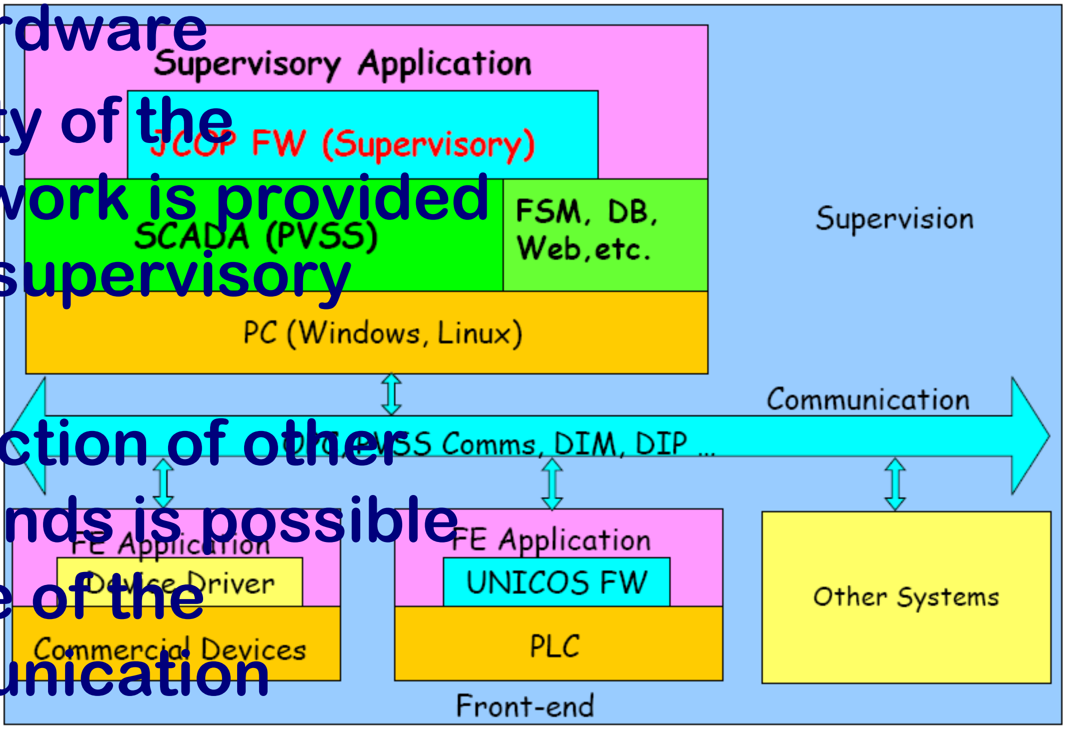
- ■ A set of guidelines and tools
  - □ Produced in collaboration, components shared
  - □ Produced using common tools
  - □ Aims for homogeneity

- ■ The framework is based on:
  - □ SCADA system – PVSS
    - ➢ Device Description (Run-time Database)
    - ➢ Device Access (OPC, Profibus, drivers)
    - ➢ Alarm Handling (Generation, Filtering, etc)
    - ➢ User Interface Builder
    - ➢ Etc …
  - □ SMI++
    - ➢ Abstract behavior modeling (Finite State Machine toolkit)

JCOP Framework

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

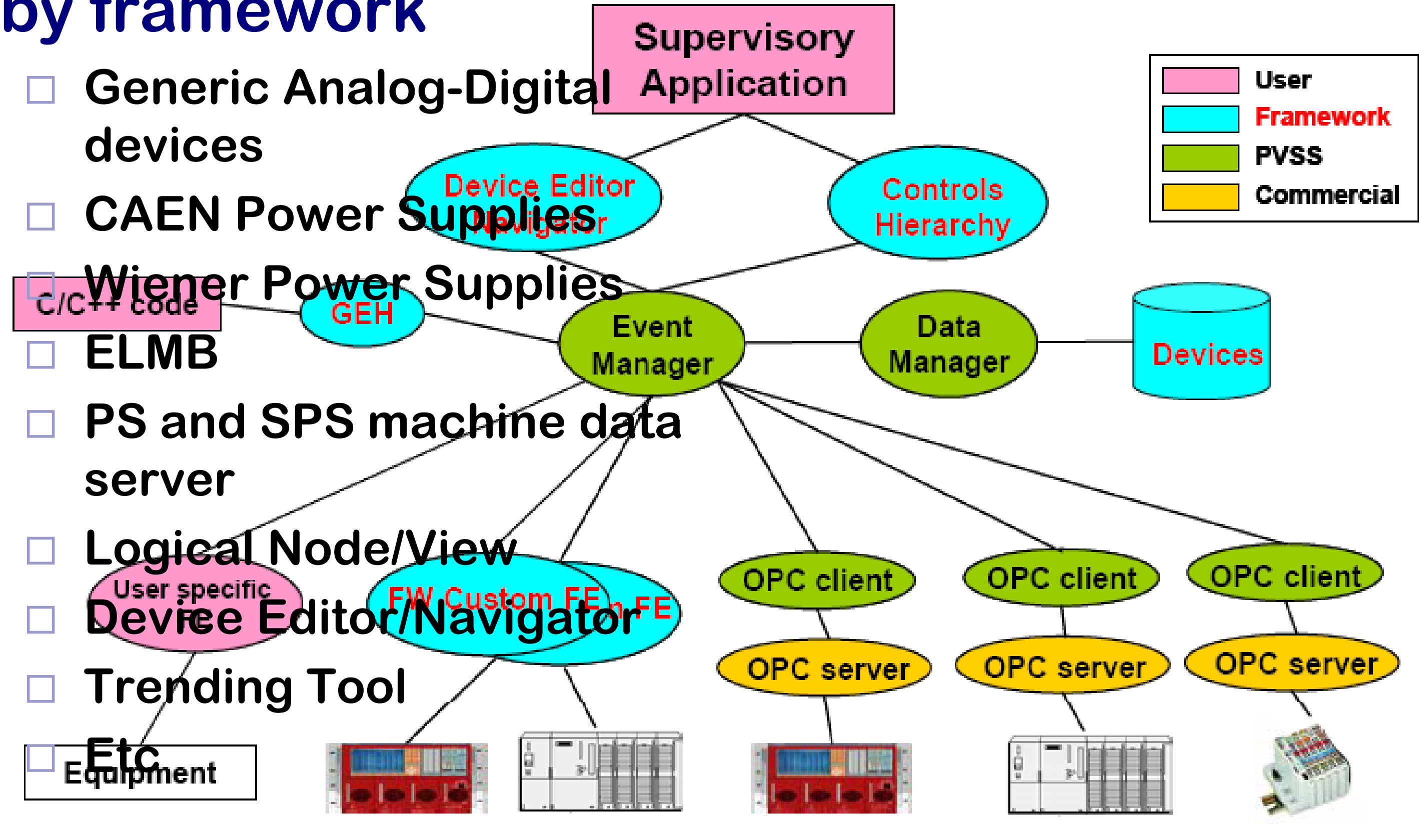
- Covers all levels down to the connection to the hardware
- Majority of the framework is provided at the supervisory level
- Connection of other front-ends is possible via one of the communication interfaces
  - □ OPC, DIM or DIP



JCOP Framework

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

- Components provided by framework
  - □ Generic Analog-Digital devices
  - □ CAEN Power Supplies
  - □ Wiener Power Supplies
  - □ ELMB
  - □ PS and SPS machine data server
  - □ Logical Node/View
  - □ Device Editor/Navigator
  - □ Trending Tool
  - □ Etc



What is FSM?

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

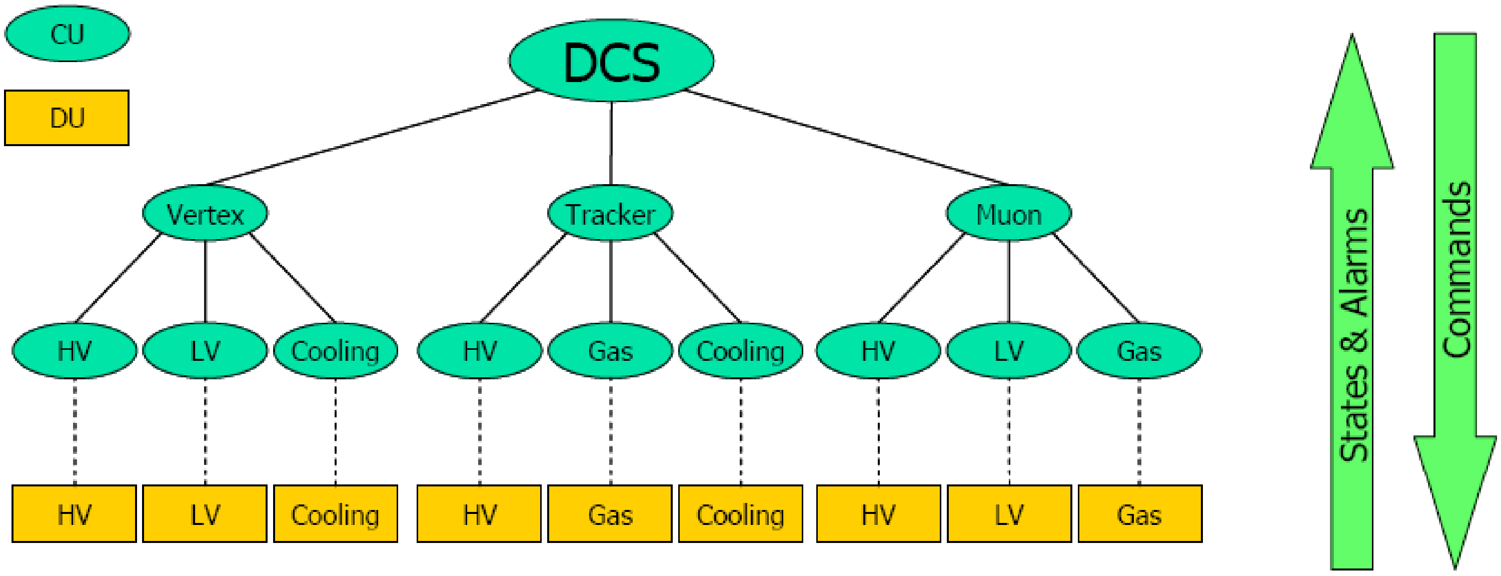
- **Abstract representation of the experiment**
  - □ Control system is modelled as a hierarchy of FSM
  - □ A generic, data-driven mechanism for modelling the functionality of a piece of equipment or a subsystem
- **A state/command interface between a parent and its children**
  - □ What *state* is it in? Is it taking data? Is it in standby? Is it broken? Is it switched off? What triggers it to move from one of these *states* to another?
- **Two types of objects are defined:**
  - □ Control Unit
    - ➢ A software entity that monitors the state of its children
  - □ Device Unit
    - ➢ A software entity that represents some hardware

FSM Model

- DCS
- PVSS/JCOP Framework
- **Finite State Machine**
- DEN
- Background and Objective
- Conclusion

A simple control system modelled using FSM

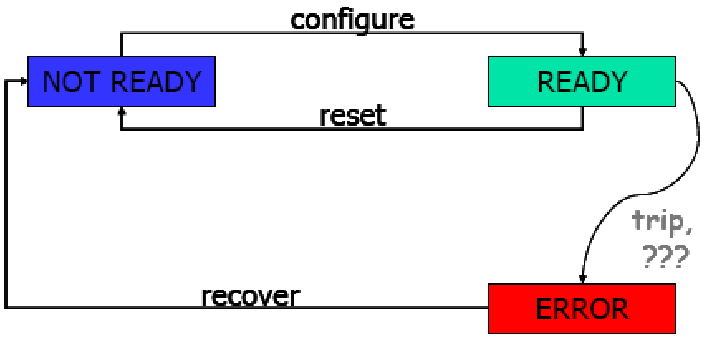
- □ Nodes have only one parent
- □ No horizontal lines
- □ Commands go down, States propagated up



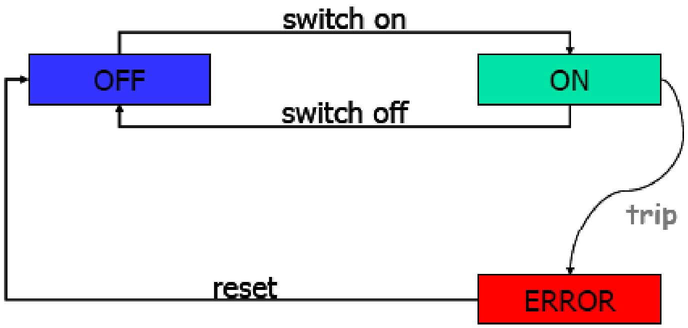
Control and Device Units

- DCS
- PVSS/JCOP Framework
- **Finite State Machine**
- DEN
- Background and Objective
- Conclusion

- State Transition Diagram for a CU



- State Transition Diagram for a DU

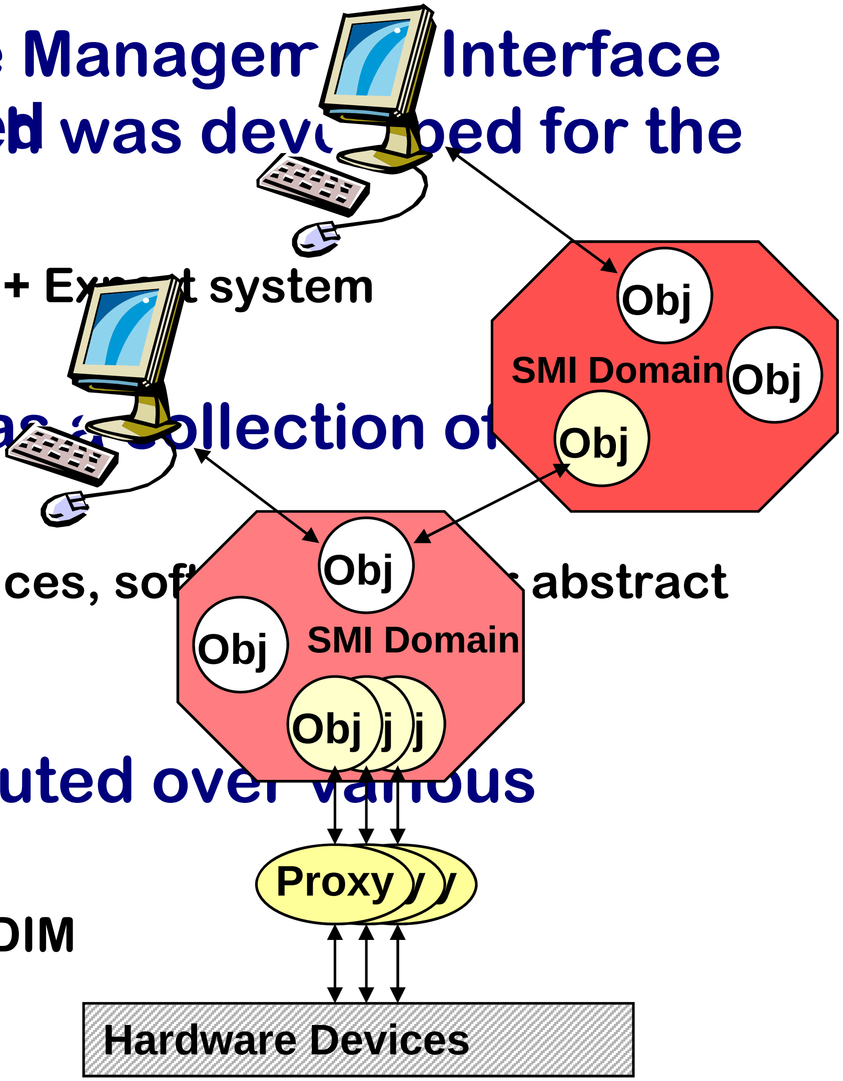


SMI ++

- DCS
- PVSS/JCOP Framework
- **Finite State Machine**
- DEN
- Background and Objective
- Conclusion

- JCOP chose the State Managem Interface (SMI++) product which was deve ped for the Delphi experiment
  - SMI = Tool to build a FMS + Ex t system
- Real world is viewed as a collection of behaving as FSM
  - Represent hardware devices, sof abstract subsystem
- SMI++ objects distributed over various platforms
  - Communication through DIM

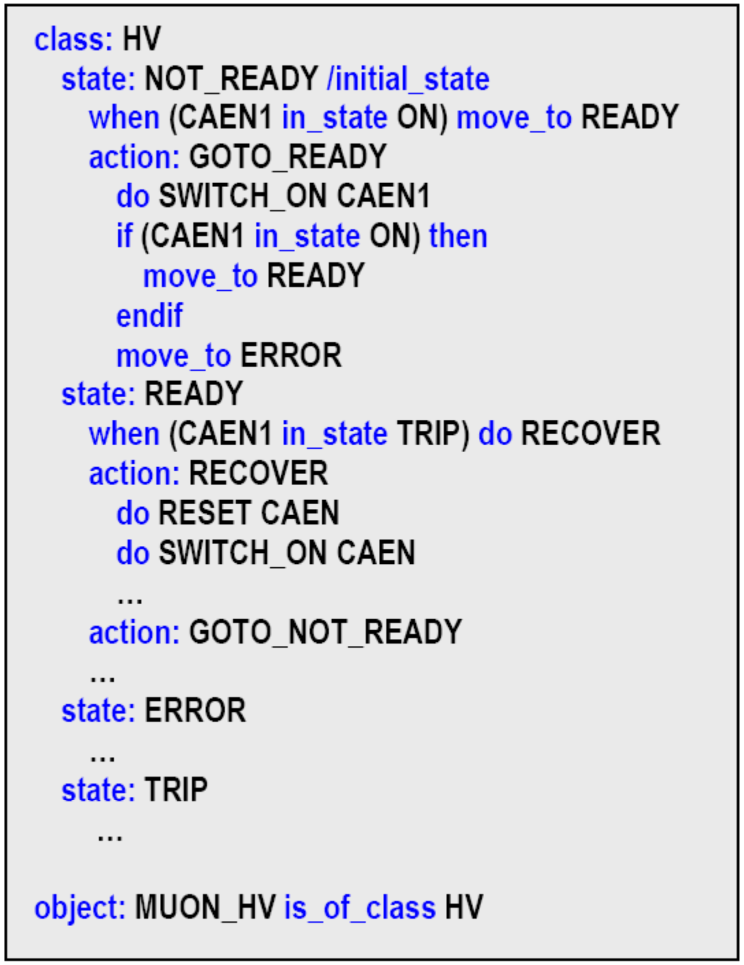
- Logically related objects can be grouped inside “SMI domains” to represent a sub-system
- Only one object (top level object) is accessed by other objects
- Important attributes: State and Action



State Management Language

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

- A special language is provided for the object description – State Management Language (SML)
- Interpreted by a Logic Engine (coded in C++) to drive the control system
- An SML file corresponds to an SMI domain



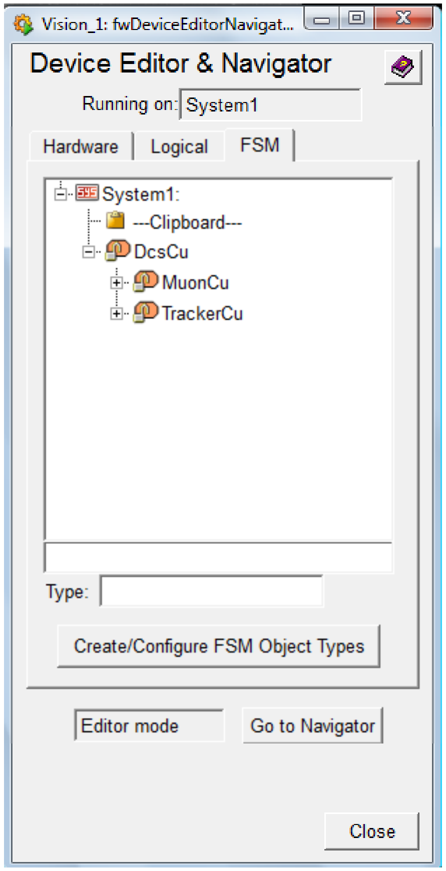
```
class: HV
  state: NOT_READY /initial_state
    when (CAEN1 in_state ON) move_to READY
    action: GOTO_READY
      do SWITCH_ON CAEN1
      if (CAEN1 in_state ON) then
        move_to READY
      endif
      move_to ERROR
  state: READY
    when (CAEN1 in_state TRIP) do RECOVER
    action: RECOVER
      do RESET CAEN
      do SWITCH_ON CAEN
      ...
    action: GOTO_NOT_READY
    ...
  state: ERROR
    ...
  state: TRIP
    ...

object: MUON_HV is_of_class HV
```

Device Editor and Navigator (DEN)

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- **DEN**
- Background and Objective
- Conclusion

- **Main interface to the Framework**
  - □ System management
- **Configuration and operation of device**
- **Editor and Navigator mode**
- **Hardware, logical and FSM view**



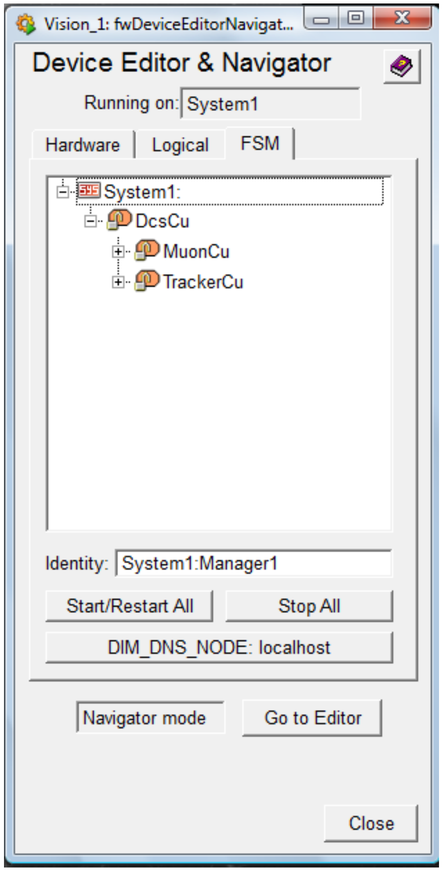
Vision_1: fwDeviceEditorNavigat...

Device Editor & Navigator

Running on: System1

Type:

Editor mode



Vision_1: fwDeviceEditorNavigat...

Device Editor & Navigator

Running on: System1

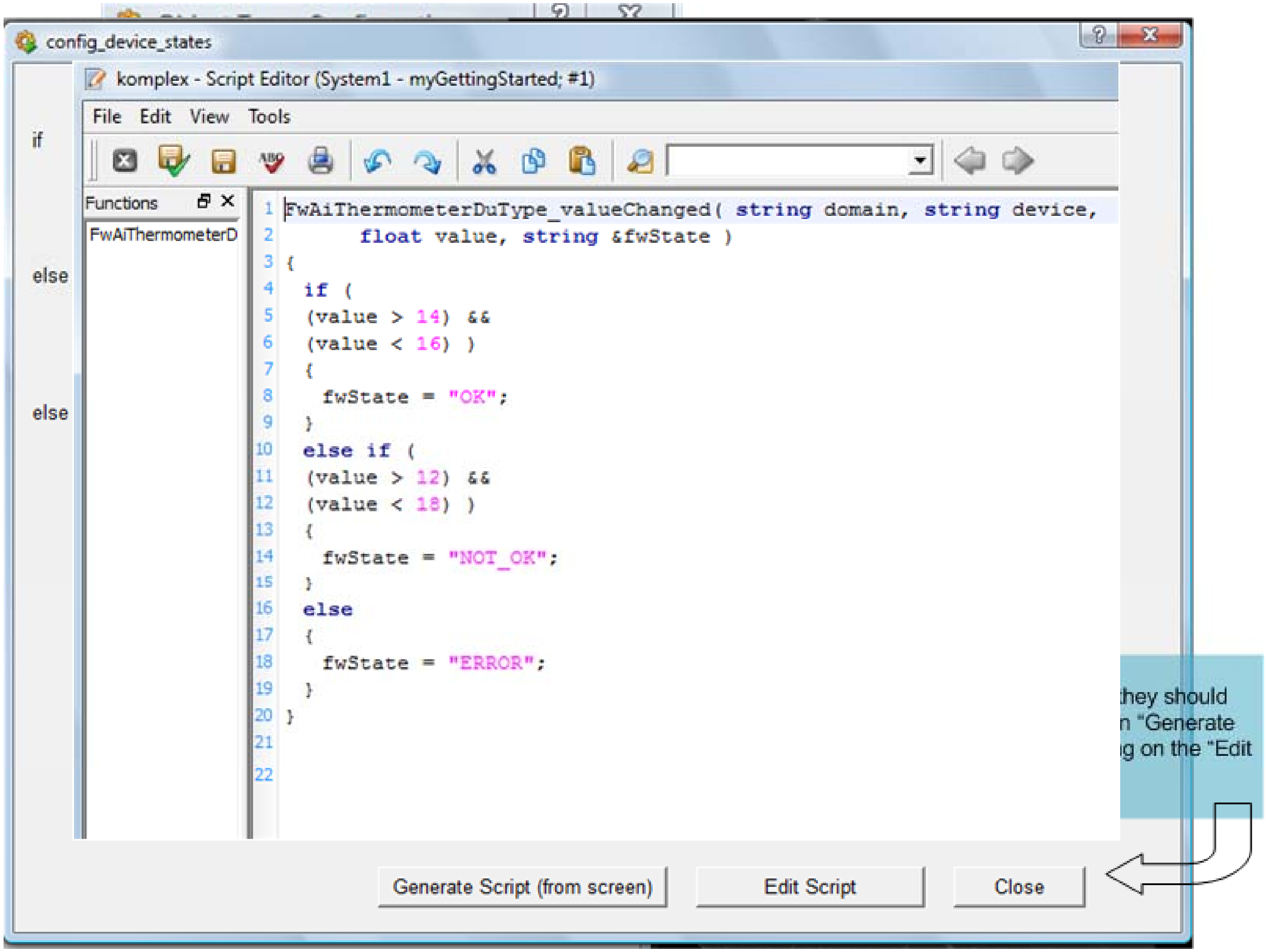
Identity: System1:Manager1

DIM_DNS_NODE: localhost

Navigator mode

Device Unit Type Configurations

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

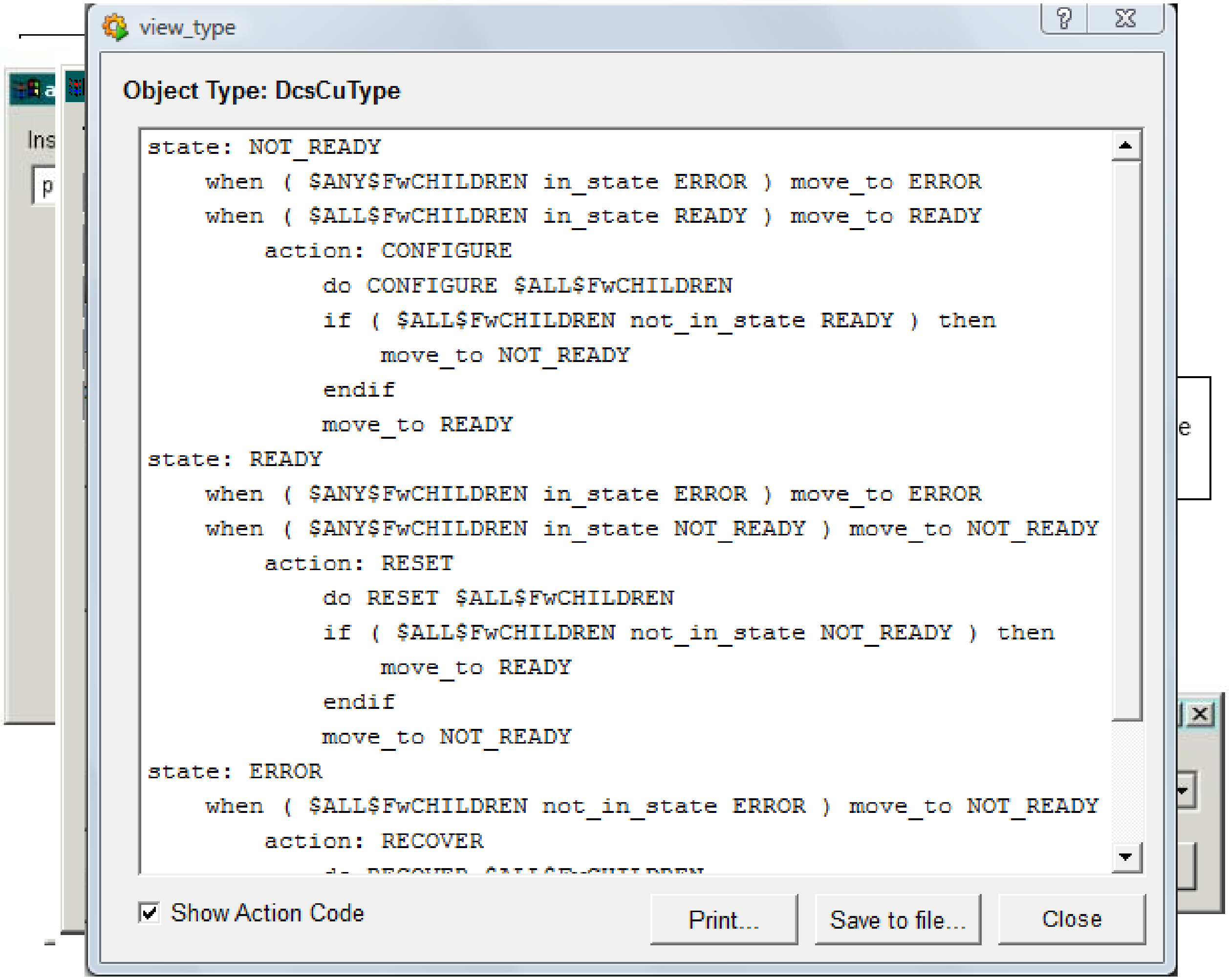


```
FwAiThermometerDuType_valueChanged( string domain, string device,
      float value, string &fwState )
{
 if (
 (value > 14) &&
 (value < 16) )
 {
  fwState = "OK";
 }
 else if (
 (value > 12) &&
 (value < 18) )
 {
  fwState = "NOT_OK";
 }
 else
 {
  fwState = "ERROR";
 }
}
```

they should
n "Generate
g on the "Edit

Logical Object Type Configuration

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion



Object Type: DcsCuType

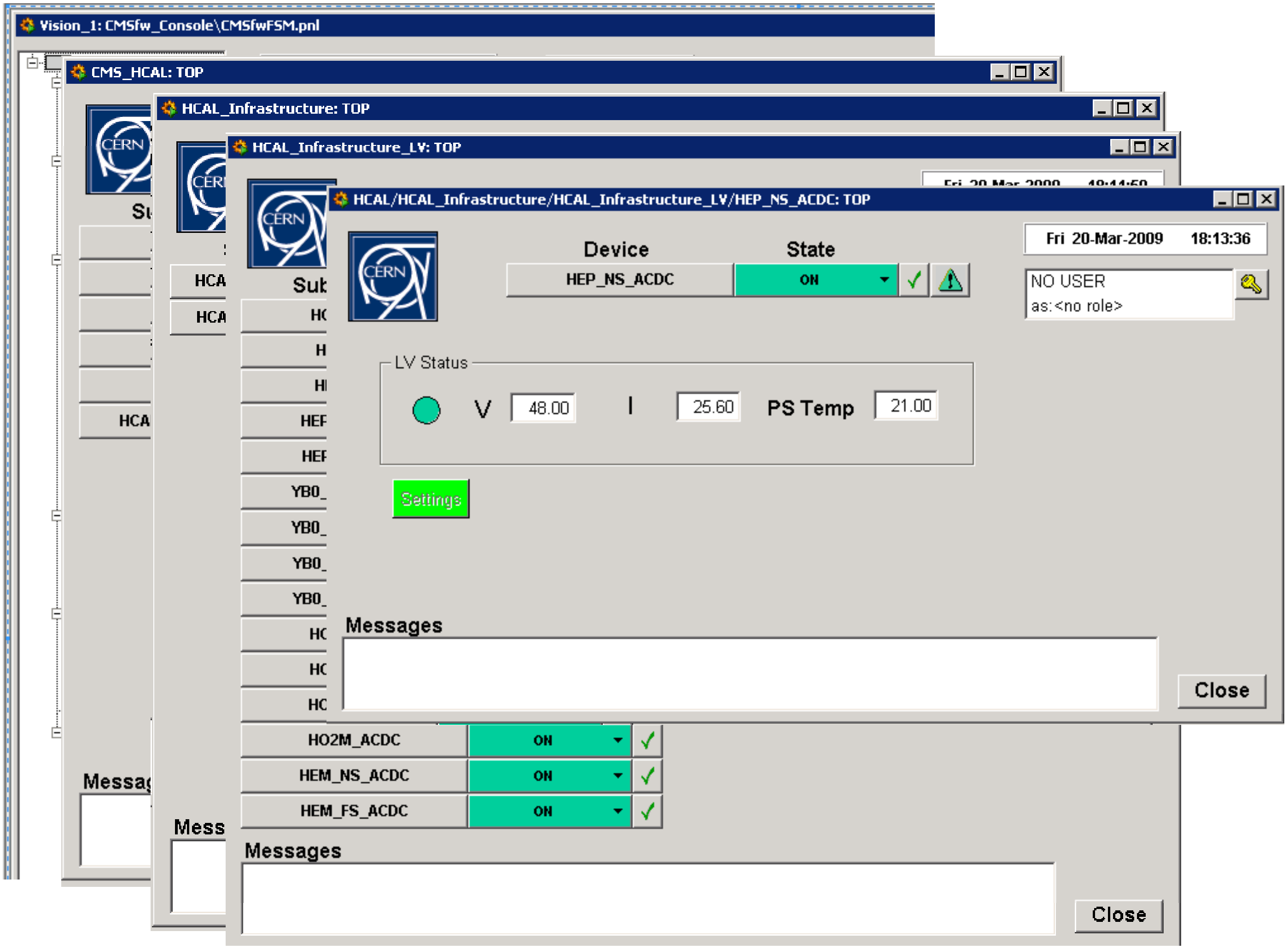
```
state: NOT_READY
    when ( $ANY$FwCHILDREN in_state ERROR ) move_to ERROR
    when ( $ALL$FwCHILDREN in_state READY ) move_to READY
        action: CONFIGURE
            do CONFIGURE $ALL$FwCHILDREN
            if ( $ALL$FwCHILDREN not_in_state READY ) then
                move_to NOT_READY
            endif
            move_to READY
state: READY
    when ( $ANY$FwCHILDREN in_state ERROR ) move_to ERROR
    when ( $ANY$FwCHILDREN in_state NOT_READY ) move_to NOT_READY
        action: RESET
            do RESET $ALL$FwCHILDREN
            if ( $ALL$FwCHILDREN not_in_state NOT_READY ) then
                move_to READY
            endif
            move_to NOT_READY
state: ERROR
    when ( $ALL$FwCHILDREN not_in_state ERROR ) move_to NOT_READY
        action: RECOVER
            do RECOVER $ALL$FwCHILDREN
```

☑ Show Action Code

Navigator Mode

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- **DEN**
- Background and Objective
- Conclusion

■ View the hierarchy starting from any control unit



Vision_1: CMSfw_Console\CMSfwFSM.pnl

CMS_HCAL: TOP

HCAL_Infrastructure: TOP

HCAL_Infrastructure_LV: TOP

HCAL/HCAL_Infrastructure/HCAL_Infrastructure_LV/HEP_NS_ACDC: TOP

Fri 20-Mar-2009 18:13:36

NO USER
as:<no role>

Device	State
HEP_NS_ACDC	ON ✓ ⚠

LV Status

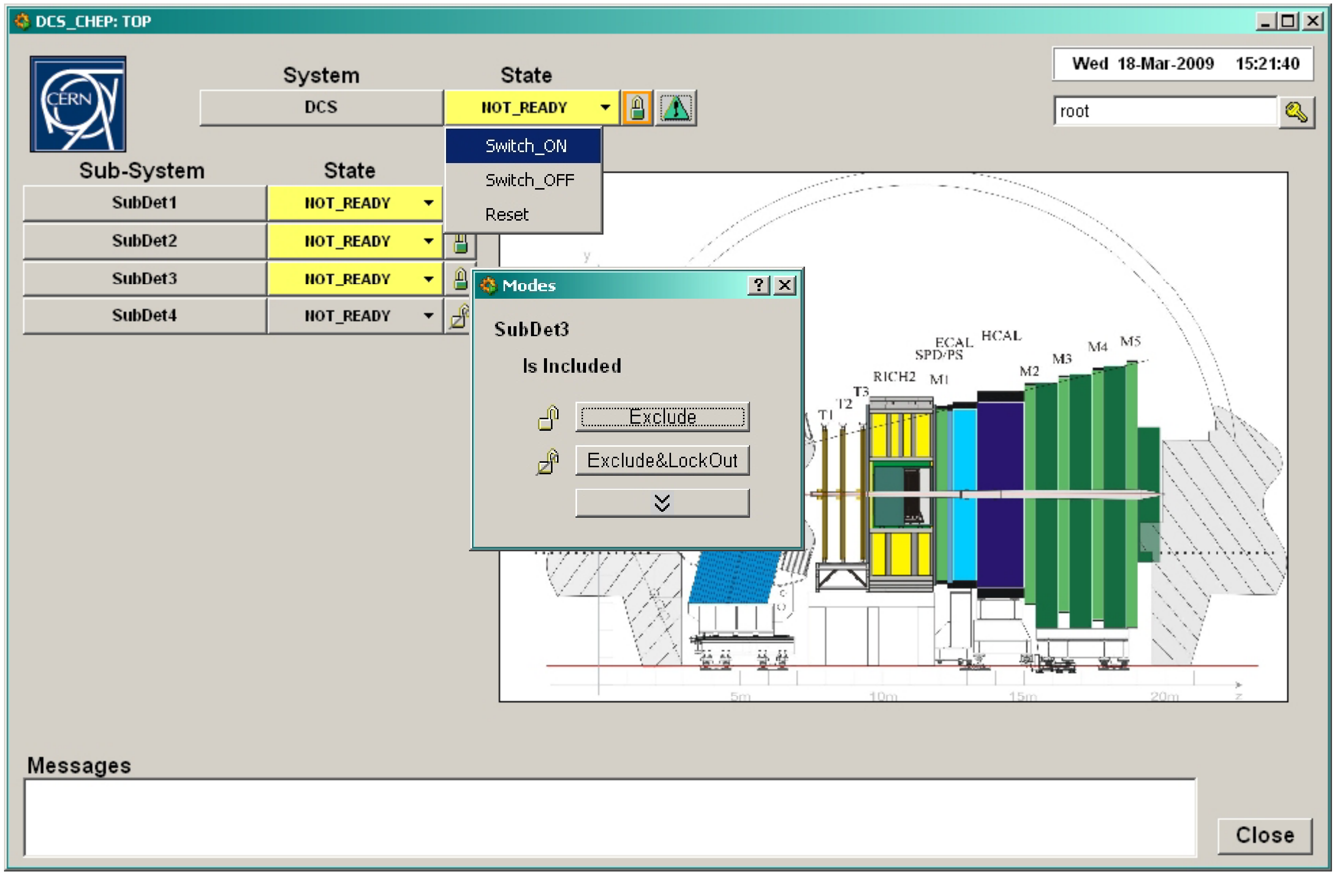
● V 48.00 I 25.60 PS Temp 21.00

HO2M_ACDC	ON ✓
HEM_NS_ACDC	ON ✓
HEM_FS_ACDC	ON ✓

Navigator Mode

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

■ Partitioning, actions etc



DCS_CHEP: TOP

Wed 18-Mar-2009 15:21:40

root

System	State
DCS	NOT_READY

Sub-System	State
SubDet1	NOT_READY
SubDet2	NOT_READY
SubDet3	NOT_READY
SubDet4	NOT_READY

SubDet3

Is Included

Messages

Background

- Background of project
  - □ Large number of independent teams
  - □ High complexity
  - □ Very different operational mode

- Common problems in developing the FSMs:
  - □ Repetitive and redundant logic
  - □ Inconsistency (endless loops)
  - □ Inhomogeneity

Objectives

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- **Background and Objective**
- Conclusion

- Understanding of DCS tools and application
- Development of an analysis tool to analyze the FSM trees in the CMS DCS
- Optimization and thorough analysis of the current FSM system
- Provide a mechanism to ensure homogeneity in the FSMs
- Development of a visualization program

Plans

- DCS
- PVSS/JCOP Framework
- Finite State Machine
- DEN
- Background and Objective
- Conclusion

- **Training courses and conferences**
  - □ Oracle SQL course (2nd Feb – 4th Feb)
  - □ PVSS/JCOP Framework training course (9th Feb – 13th Feb)
  - □ FSM course (17th Feb – 19th Feb)
  - □ General and professional French course (26th Jan – 3rd April)
  - □ Programming in JAVA (27th April – 29th April)
  - □ CHEP (23rd Mar – 27th Mar)
  - □ RT2009 (10th May – 15th May)
  - □ ICALEPCS (12th Oct – 16th Oct) : Abstract submitted
- **Exportation of FSM into Database**
- **Analysis and building of the complex FSM trees**
- **Possible collaboration with Eindhoven University of Technology**
  - □ Proposal written and meeting planned

Conclusion

- The Detector Control System is an integral part of the CMS experiment and is large and complex
- A homogenous and coherent control system has been developed using the JCOP framework
- PVSS and SMI++ are being used in the framework, and its tools and applications have been proven to be useful and user-intuitive
- However, inhomogeneity and incoherence in the FSM development still exist, an analysis tool is foreseen to be developed and implemented

- **Thank you for your attention!**