



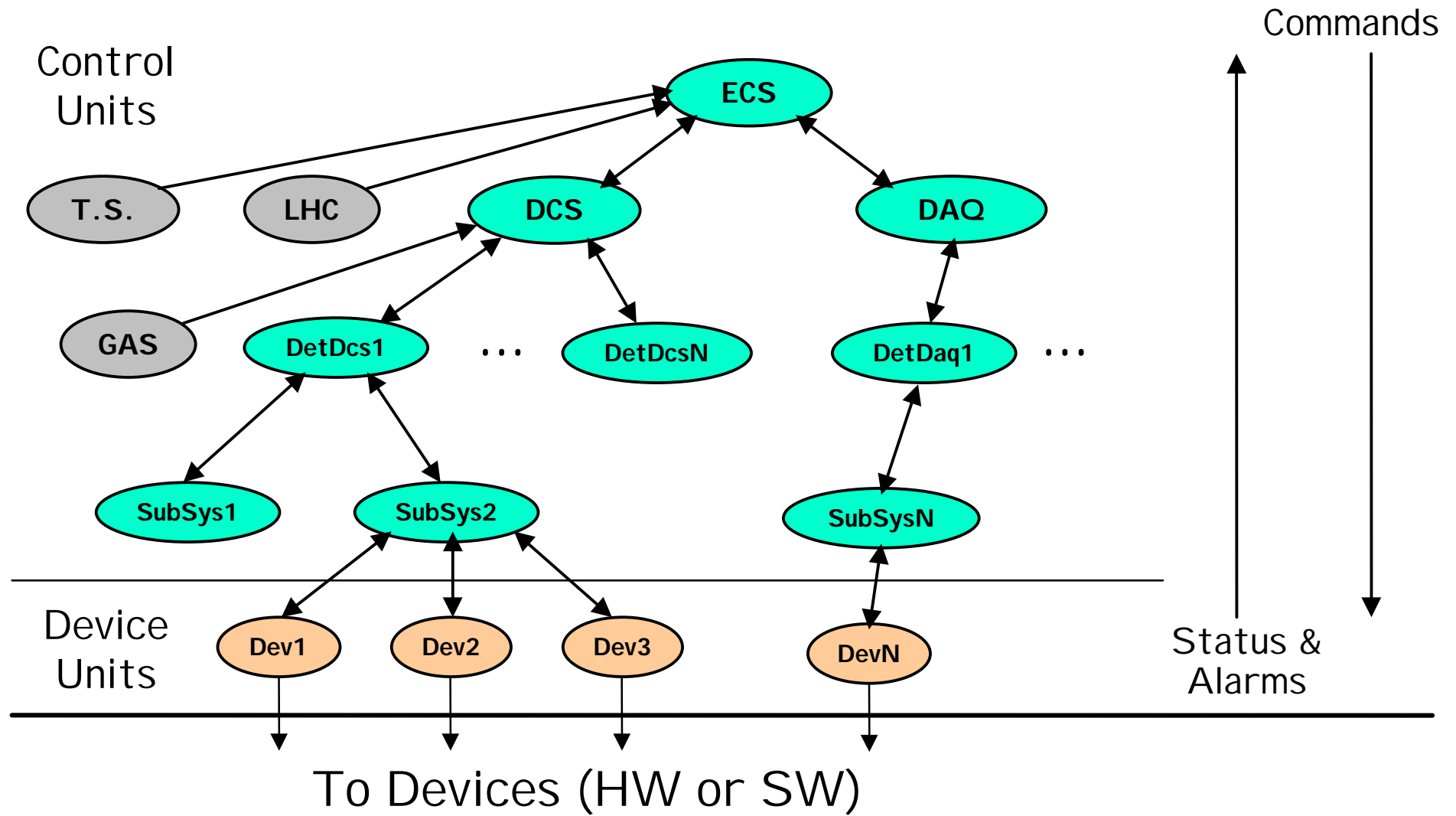
# Experiment Control System

## Architecture and Tools

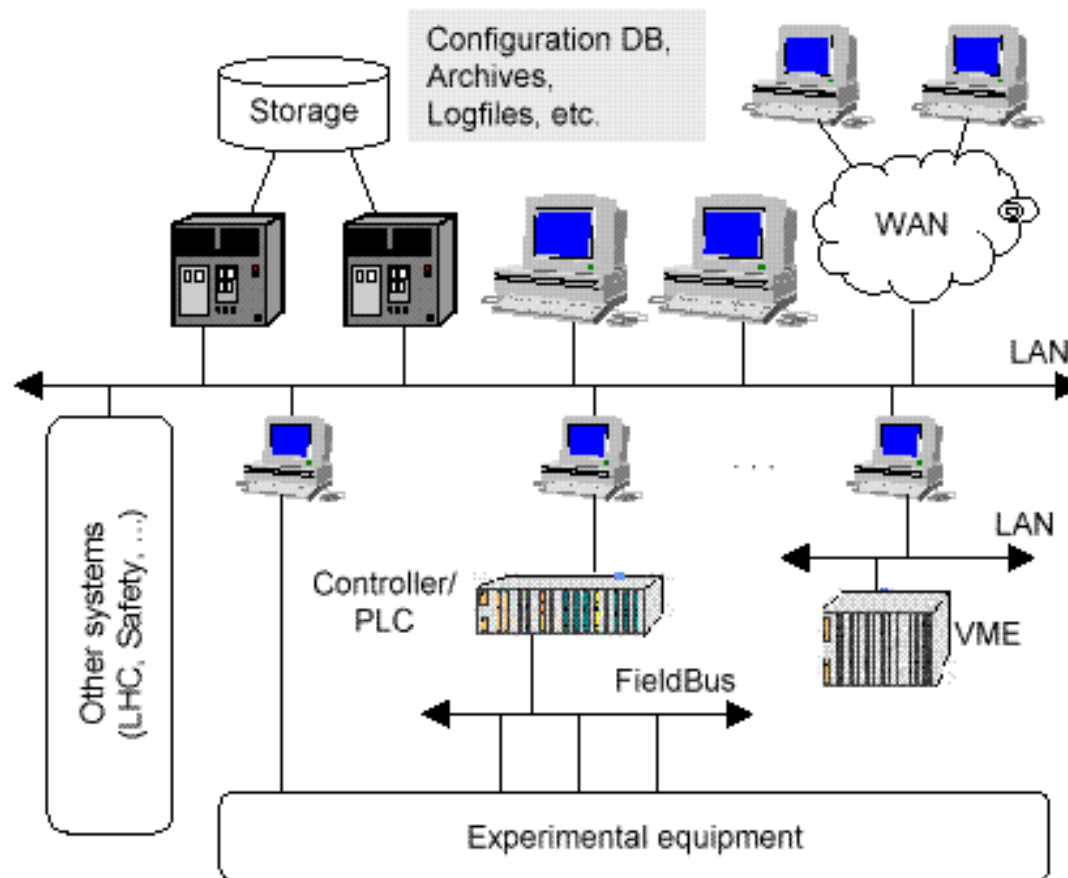
*Clara Gaspar, May 2000*



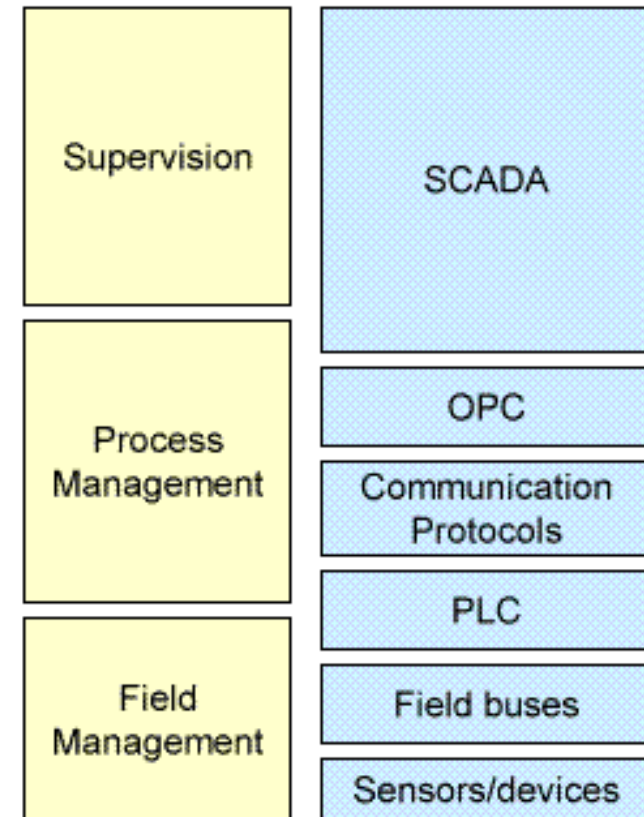
# Generic Architecture



# HW Architecture



## Technologies



SCADA = supervisory control and data acquisition  
 OPC = OLE for process control  
 PLC = Programmable logic controller  
 Field buses = CAN, ProfiBus, WorldFip, ...

# The Control Framework

---

- Will provide guidelines and tools for the implementation of all components in the tree.
- Based on:
  - A Commercial Tool - SCADA  
(tender out next week)
  - +Additions (home made or commercial)
    - | Finite State Machine Toolkit
    - | Specific drivers
    - | etc.

Dev

Tools for Device Unit implementation:

- | Device Description
- | Several Access Protocols
- | Alarm Generation Configuration
- | User Interface Editor
- | Behaviour Model
- | Predefined (Configurable) Device Components, like:
  - Power supply (CAEN, Lecroy, ...)
  - CC-PC interface
  - etc.



# Device Configuration

vision\_1: data point parametrization

File Panel ?

periphery

DPE: System1:crate0.Status[2]

driver type

- PROFIBUS FMS
- PROFIBUS DP
- SSI
- RK512
- SIM
- ABB
- SEBA
- ADS
- MUEHBUS
- OPCClient
- DIMClient
- DIMServer

parametrize ...

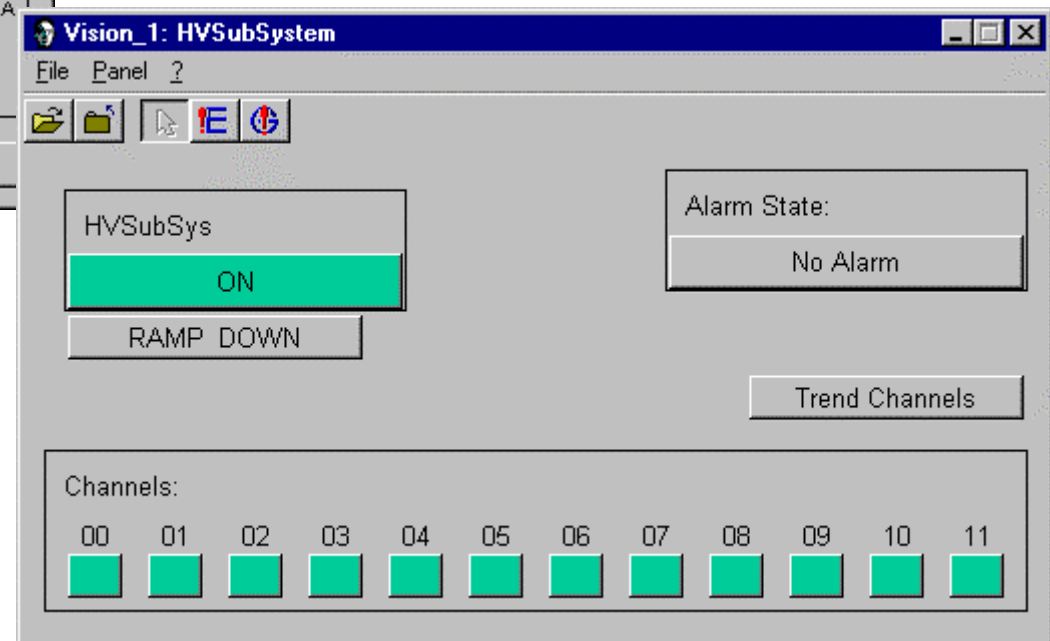
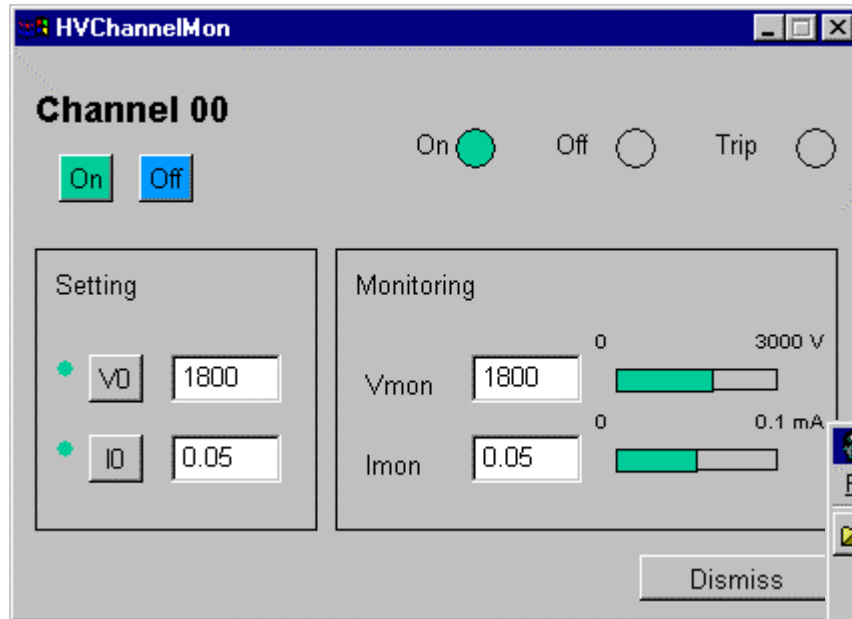
close

Tree view:

- FwCtrlUnitUi
- HVCAENcrate
  - crate0
    - alert\_class
    - common
    - lock
    - Do
      - common
      - lock
    - ClrAlm
    - Status
      - common
      - lock
    - ALRM
      - original
      - address
      - alert\_hdl
      - archive
      - common
      - lock
    - Activel
    - ActiveV
    - HVENBL
    - KbdENBL
    - PWD
    - PWRStat
  - Channels
- HVCAENSY127Channel
- HVChannel
- HVCrate



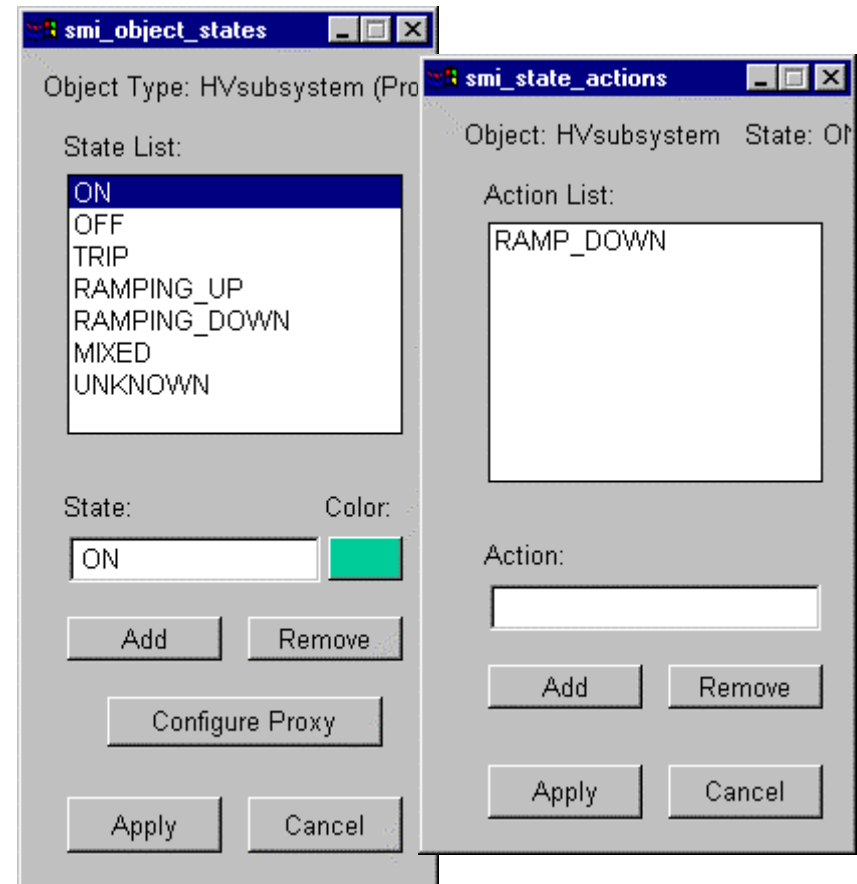
# Device User Interfaces





# Device Behaviour Model

- User tasks:  
(with the help of tools)
  - Define Device States
  - Define Actions possible in each state
  - Derive State from Device status "bits"
  - Implement Action by setting the relevant "bits".

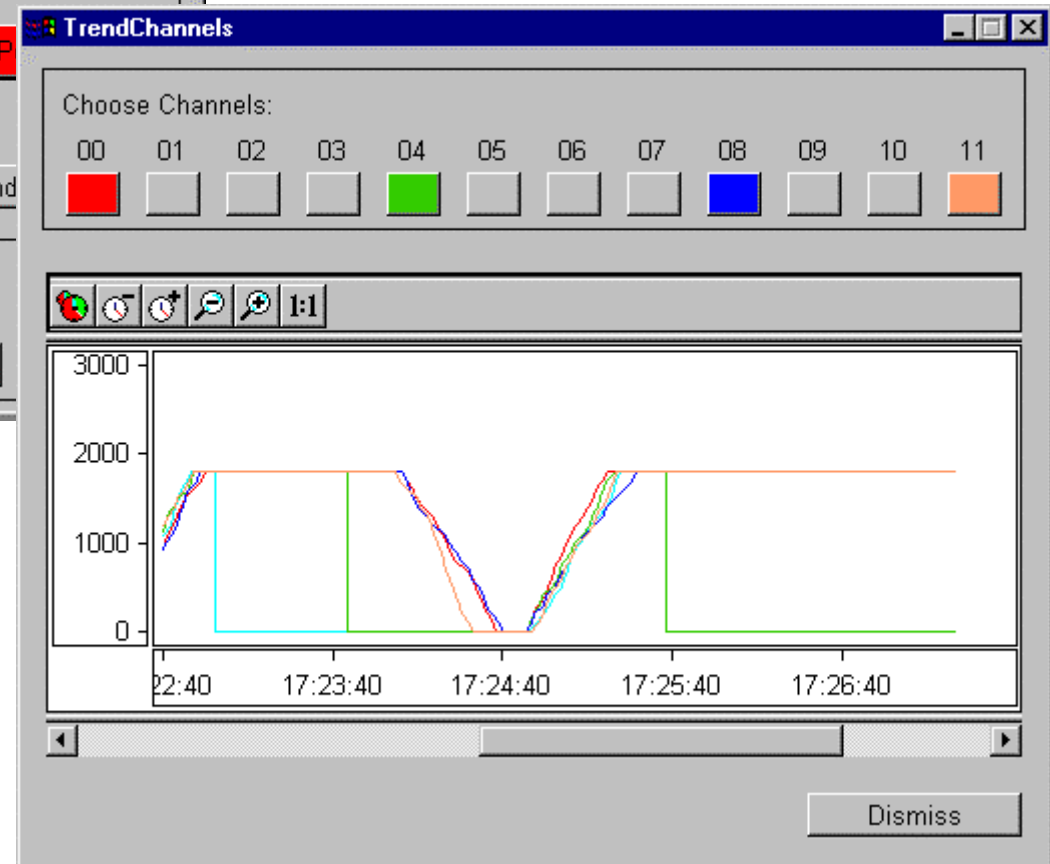






# Predefined Components

The screenshot shows the 'Vision\_1: HVSubSystem' window. It features a menu bar with 'File' and 'Panel ?'. Below the menu is a toolbar with icons for file operations and zooming. The main area contains a 'HVSUBSYS' section with a red 'TRIP' indicator and a 'CLEAR TRIP' button. To the right, an 'Alarm State:' section shows 'Alarm P'. At the bottom, a 'Channels:' section displays ten colored indicators (00-09), with channel 04 highlighted in red.



# Control Framework (cont.)

DCS

## Tools for Developing the Control Units:

### | Control Unit Configuration

- Which Components: Devices and/or Control Units

### | Logic Behaviour Modeling (FSM)

- Model the dependencies between components
- Automate Operations & Error Recovery

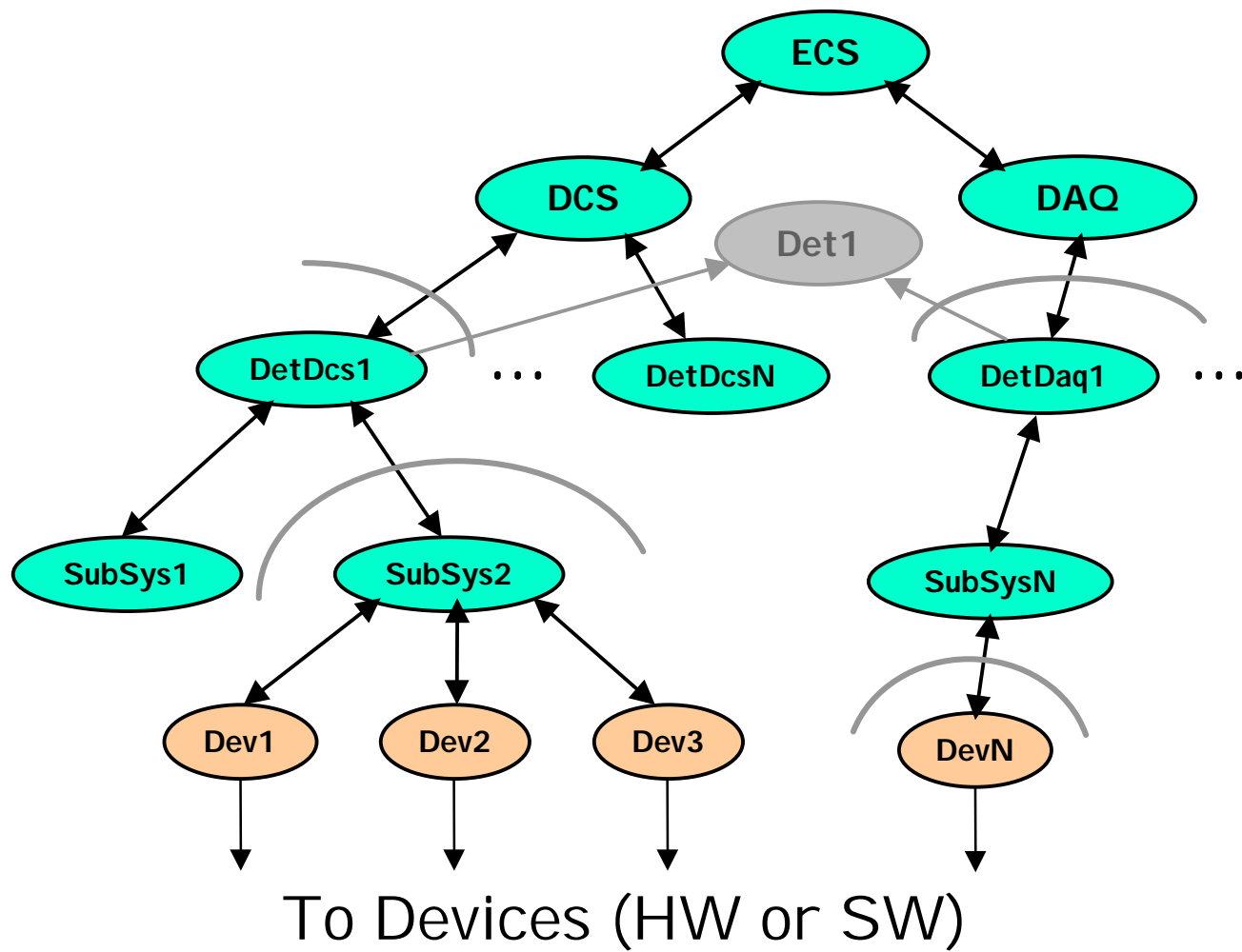
### | Alarm Handling

- Filtering, Summarising, Displaying, Masking, etc

### | Partitioning

### | User Interface Generation

# Partitioning



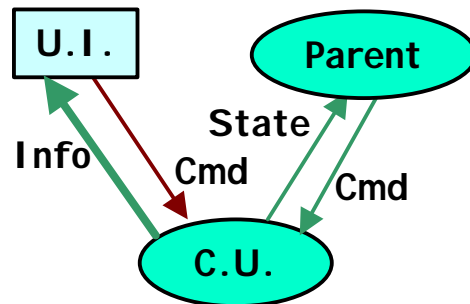
# Control Unit Operation Modes

## Control Units:

- Publish State and Information
- Can receive commands

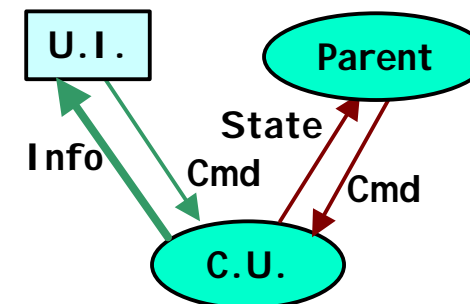
### Normal Operation

- Hierarchical control only



### Partitioned

- No Hierarchical control
- Control from a "local" U.I.



# Control Unit tasks

---

- | Each C.U. is “inherently” capable of:
  - | Publishing information/receiving commands
  - | Handling Partitioning
  - | Communicating with its children
    - Send Commands / Receive states
  - | Take decisions based on received states  
(Logic behaviour described using graphic tools)
  - | Filtering and Summarizing Alarms
  - | Interfacing to the user  
(U.I. Automatically generated by the tools)
    - Generic Control
    - Alarm Display ( Acknowledgment and Masking of alarms)
  - | Archiving, Retrieving and Displaying (trending) all data



# Modeling

The screenshot displays four windows from the SMI modeling software:

- smi\_domain:** Shows Domain: Det1 and Object List: Dev1, Dev2, Det1. Object Type: DetDCS1 and Object Name: Det1.
- smi\_object\_states:** Shows Object Type: DetDCS1 and State List: READY, NOT\_READY. State: READY.
- smi\_state\_when\_acti...:** Shows Object: DetDCS1 State: RE... and Action List: SET\_NOT\_READY, NV\_GOTO\_NOT\_READY. Action: SET\_NOT\_READY.
- action\_editor:** Shows Instructions: do SWITCH\_OFF Dev1, do SWITCH\_OFF Dev2, if (( Dev1 in\_state OFF ) and ( Dev2 in\_state OFF )) then terminate\_action/state = NOT\_READY, endif terminate\_action/state = READY.
- instr\_do:** Shows Send to Object: Dev1 and Action: SWITCH\_ON, SWITCH\_OFF, SWITCH\_OFF.



# Generated User Interfaces

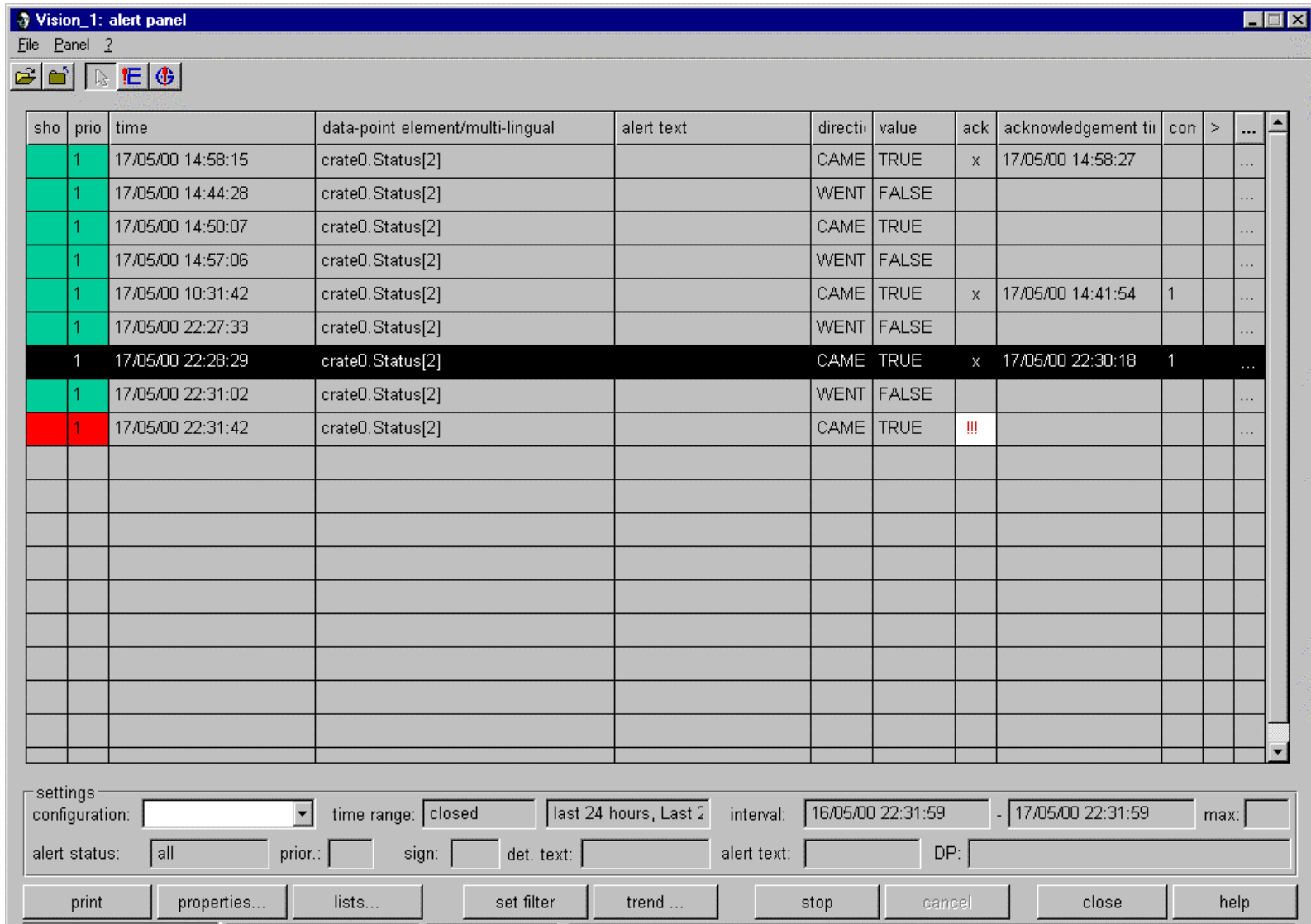
The image shows two windows from a generated user interface. The background window, titled "Vision\_1: ctrlunit\ fsm \ smiDomain.pnl", features a menu bar with "File" and "Panel", a toolbar with icons for file operations and help, and a "Select Domain:" dropdown menu currently set to "Det1". Below this, there are three control panels: one for "Det1" with a "READY" status indicator and a "SET NOT READY" button; one for "Dev1" with an "ON" status indicator; and one for "Dev2" with an "ON" status indicator.

The foreground window, titled "System1:fwUiDCS: ctrlunit\ui\main.pnl", displays a "Detector Control" window. At the top, it shows "Detector Control" in a text field and a green "READY" status indicator. Below this is a table with two columns: "Sub-systems" and "Status".

Sub-systems	Status
Sub Detector 1	READY
Sub Detector 2	READY
Sub Detector 3	READY

At the bottom of the foreground window, there is an "Action list" section containing two buttons: "SET\_NOT\_READY" and "Previous".

# Alarm Display



sho	prio	time	data-point element/multi-lingual	alert text	directi	value	ack	acknowledgement ti	com	>	...
	1	17/05/00 14:58:15	crate0.Status[2]		CAME	TRUE	x	17/05/00 14:58:27			...
	1	17/05/00 14:44:28	crate0.Status[2]		WENT	FALSE					...
	1	17/05/00 14:50:07	crate0.Status[2]		CAME	TRUE					...
	1	17/05/00 14:57:06	crate0.Status[2]		WENT	FALSE					...
	1	17/05/00 10:31:42	crate0.Status[2]		CAME	TRUE	x	17/05/00 14:41:54	1		...
	1	17/05/00 22:27:33	crate0.Status[2]		WENT	FALSE					...
	1	17/05/00 22:28:29	crate0.Status[2]		CAME	TRUE	x	17/05/00 22:30:18	1		...
	1	17/05/00 22:31:02	crate0.Status[2]		WENT	FALSE					...
	1	17/05/00 22:31:42	crate0.Status[2]		CAME	TRUE	!!!				...

settings  
configuration: [dropdown] time range: closed last 24 hours, Last 2 interval: 16/05/00 22:31:59 - 17/05/00 22:31:59 max: [input]  
alert status: all prior.: [input] sign: [input] det. text: [input] alert text: [input] DP: [input]

print properties... lists... set filter trend ... stop cancel close help



# Control Framework

---

## ■ Other Items that will be supported:

### ■ CERN recommended Fieldbuses

- | CAN, Profibus, WorldFip

### ■ Ethernet (and Credit Card PC)

- | OPC, DIM

### ■ Experiment Infrastructure

- | Rack and Crate Control, GAS Systems (GAS WG), Cooling (?)

### ■ CERN Infrastructure (Data Interchange WG)

- | Technical Services, LHC machine, LHCb Magnet(?)



# Conclusions

- | The best way to achieve an homogeneous and maintainable control system (and to save manpower) is:
  - | To do the maximum in common
    - The Controls Framework is to be developed and used by the 4 LHC experiments
    - New “Devices” should be developed in a re-usable way and included in the Framework
  - | To Standardize on HW choices as much as possible
    - So that common SW can be used
- | Please contact us for HW choices (of potentially common items)
  - | like: power supplies, Temperature Sensors, etc