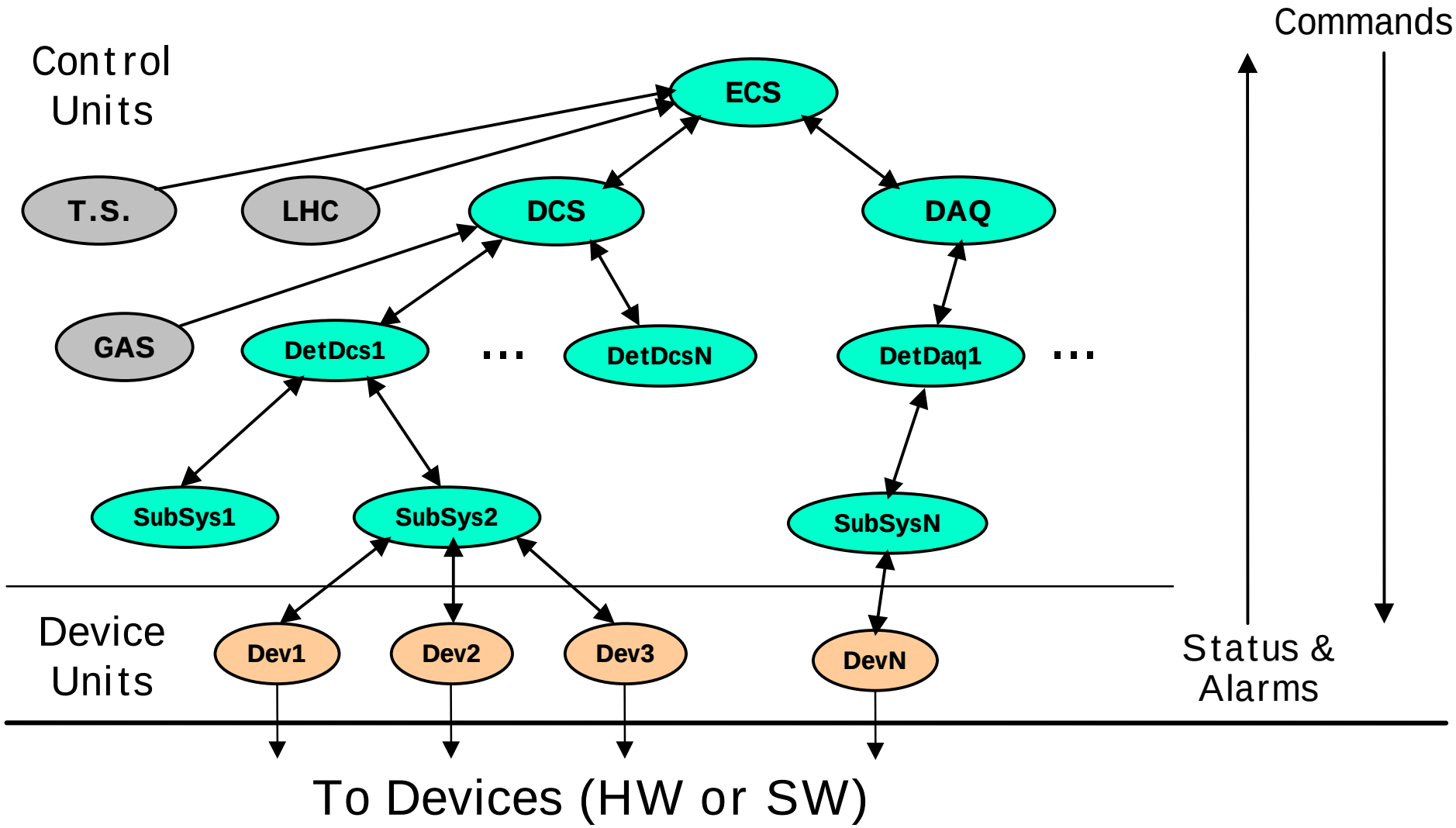
Experiment Control System

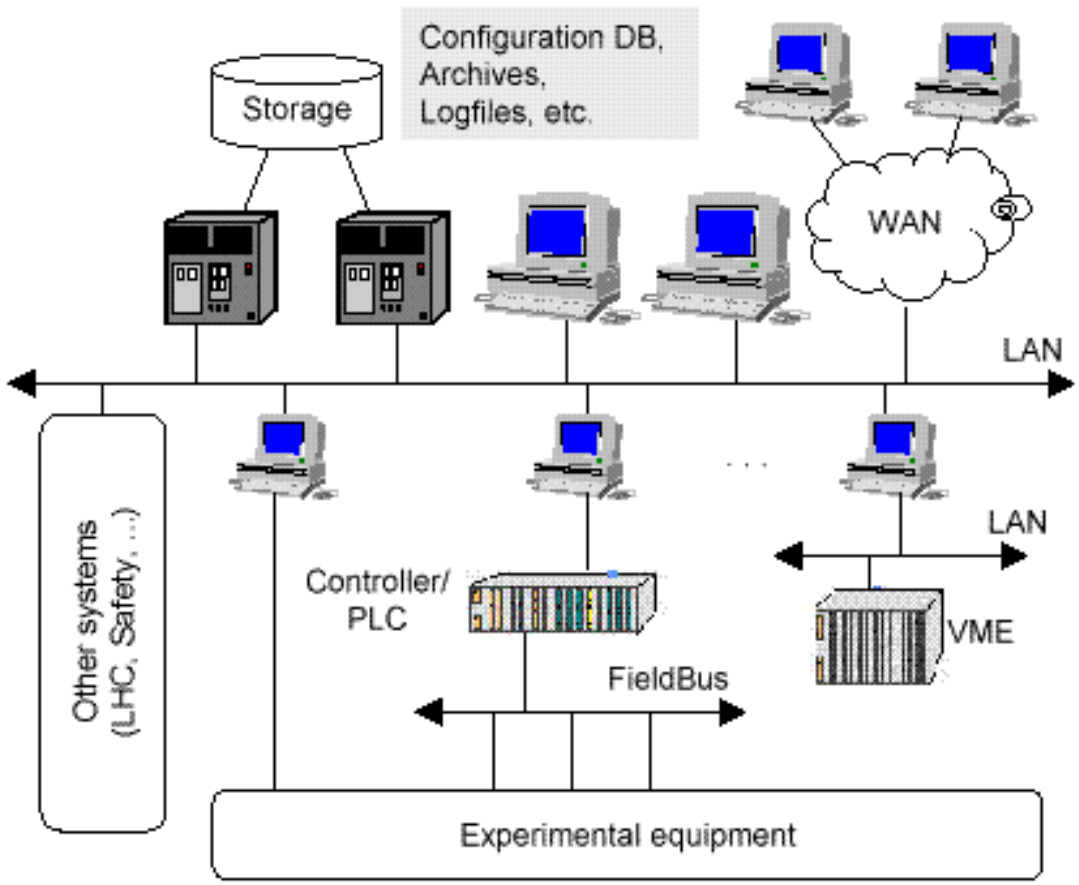
Architecture and Tools

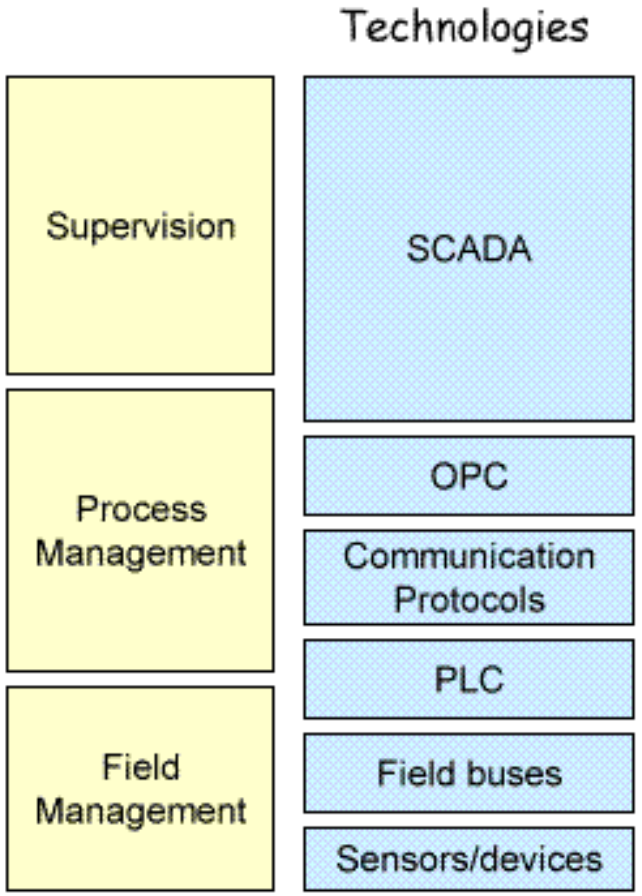
Clara Gaspar, May 2000

Generic Architecture



HW Architecture





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.

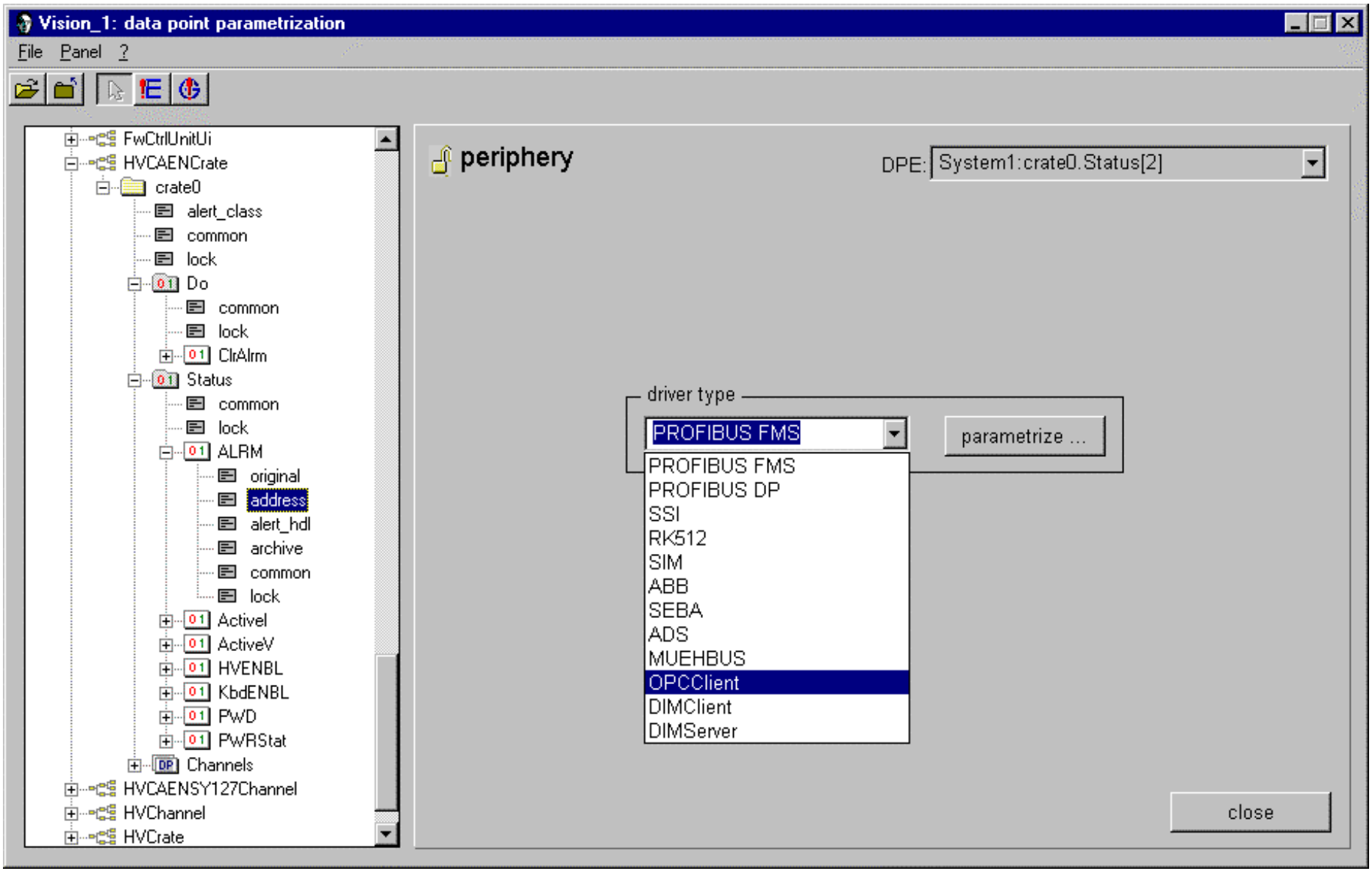
Control Framework

Dev

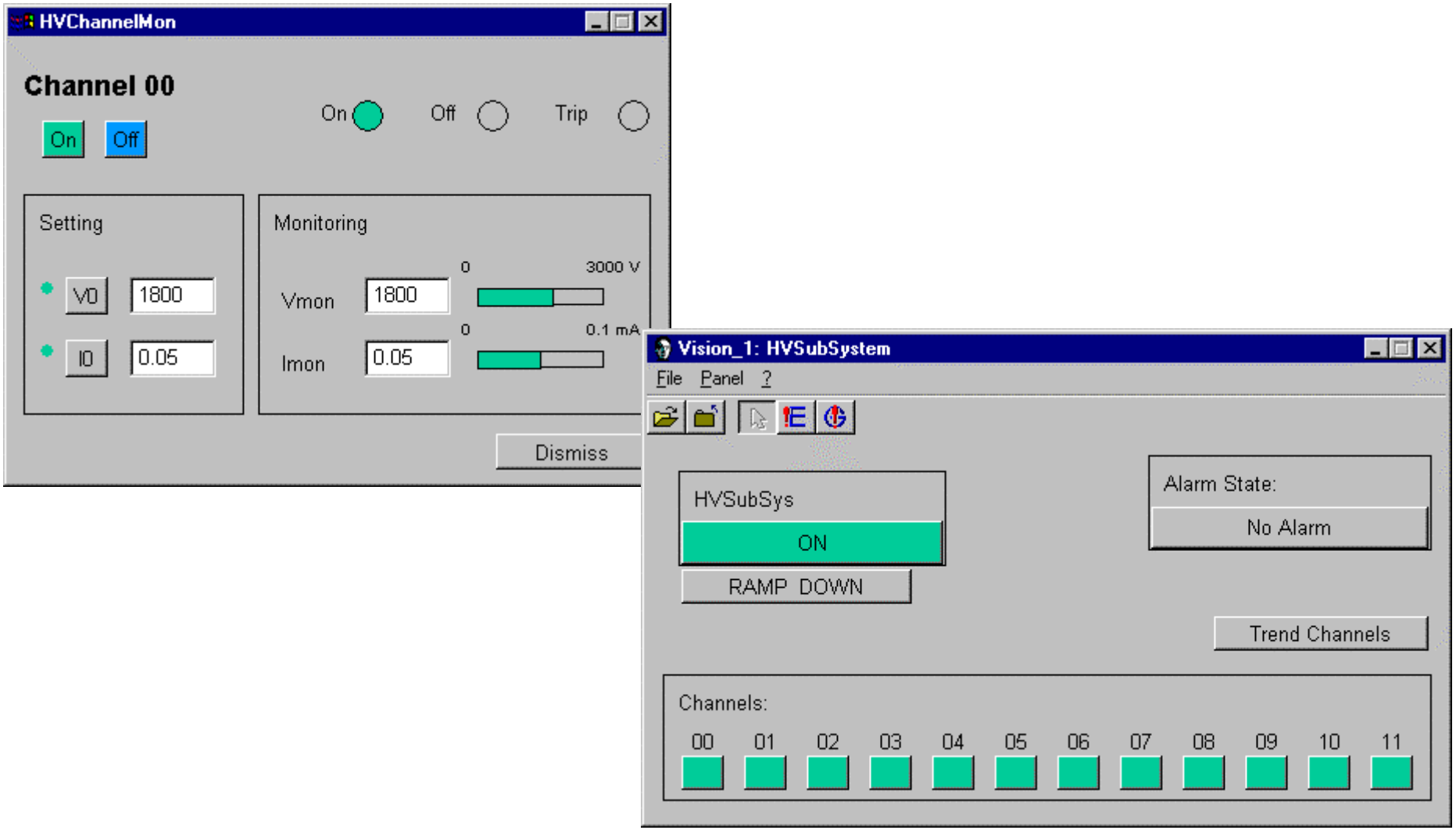
Tools for Device Unit implementation:

- 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



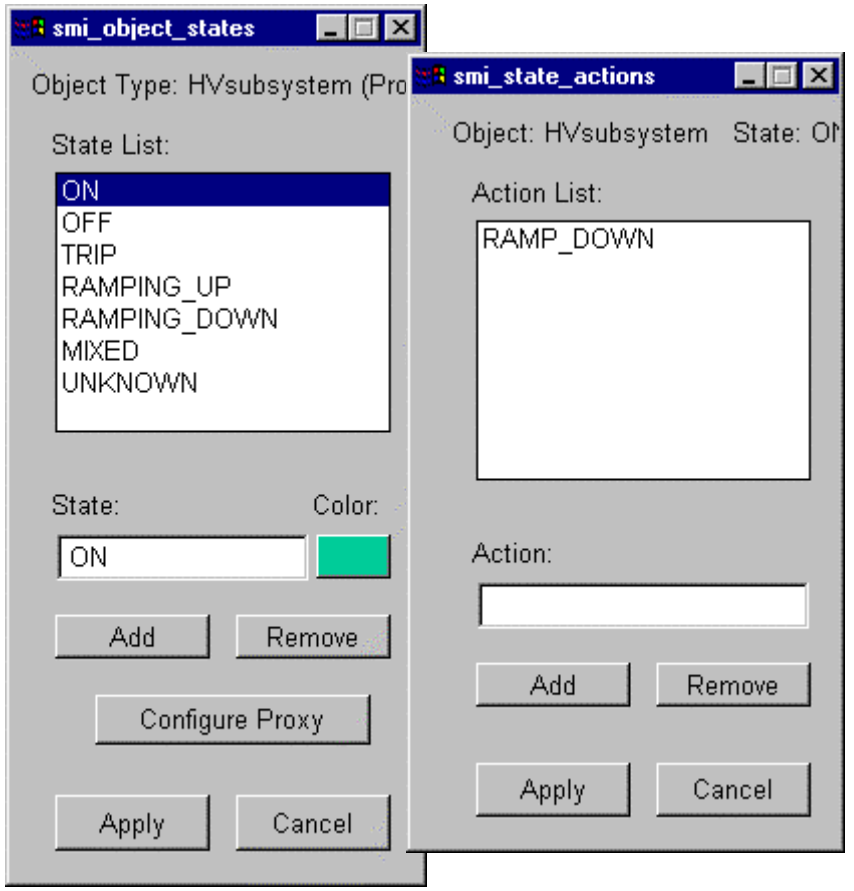
Device User Interfaces



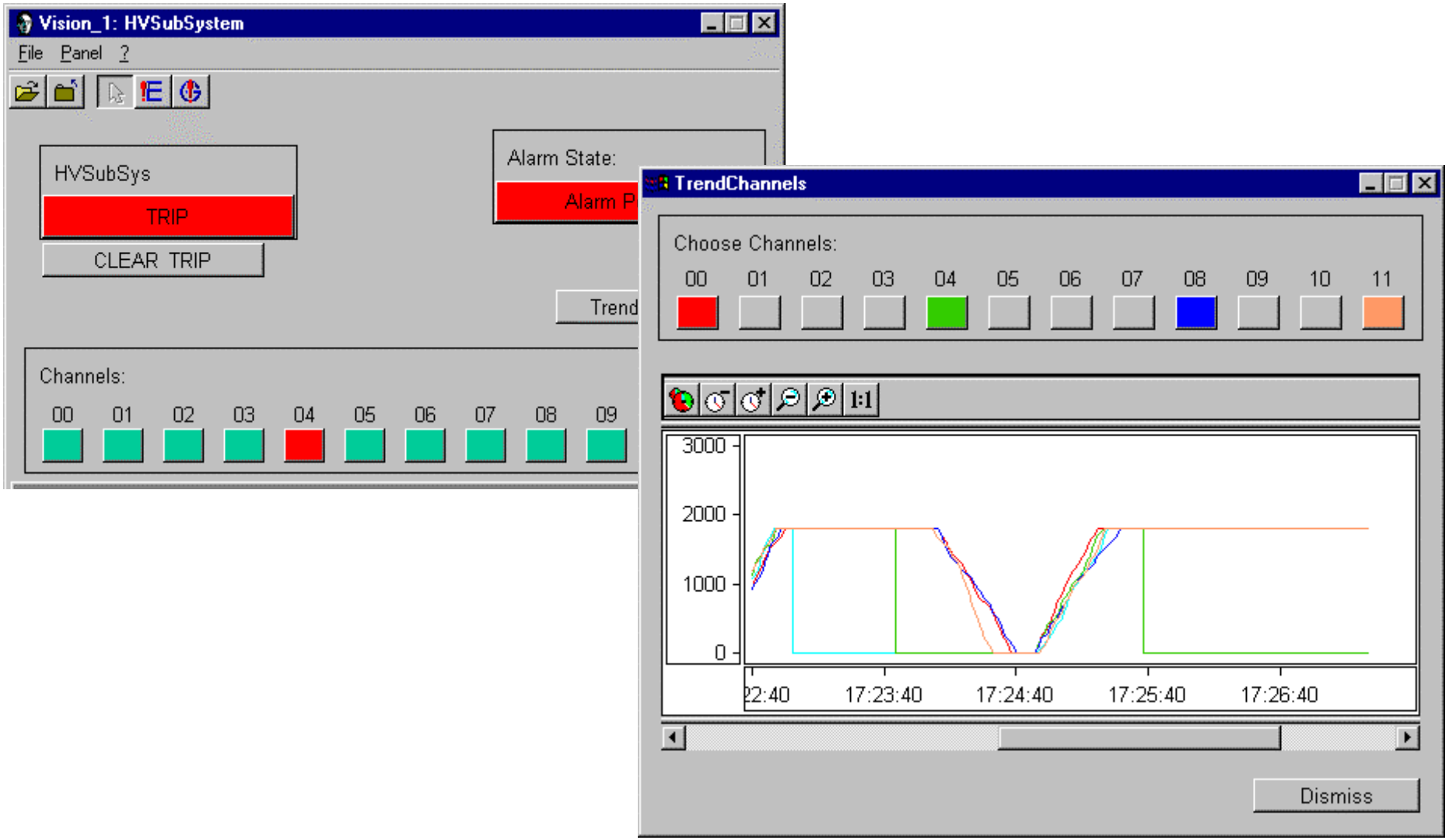
Device Behaviour Model

- 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



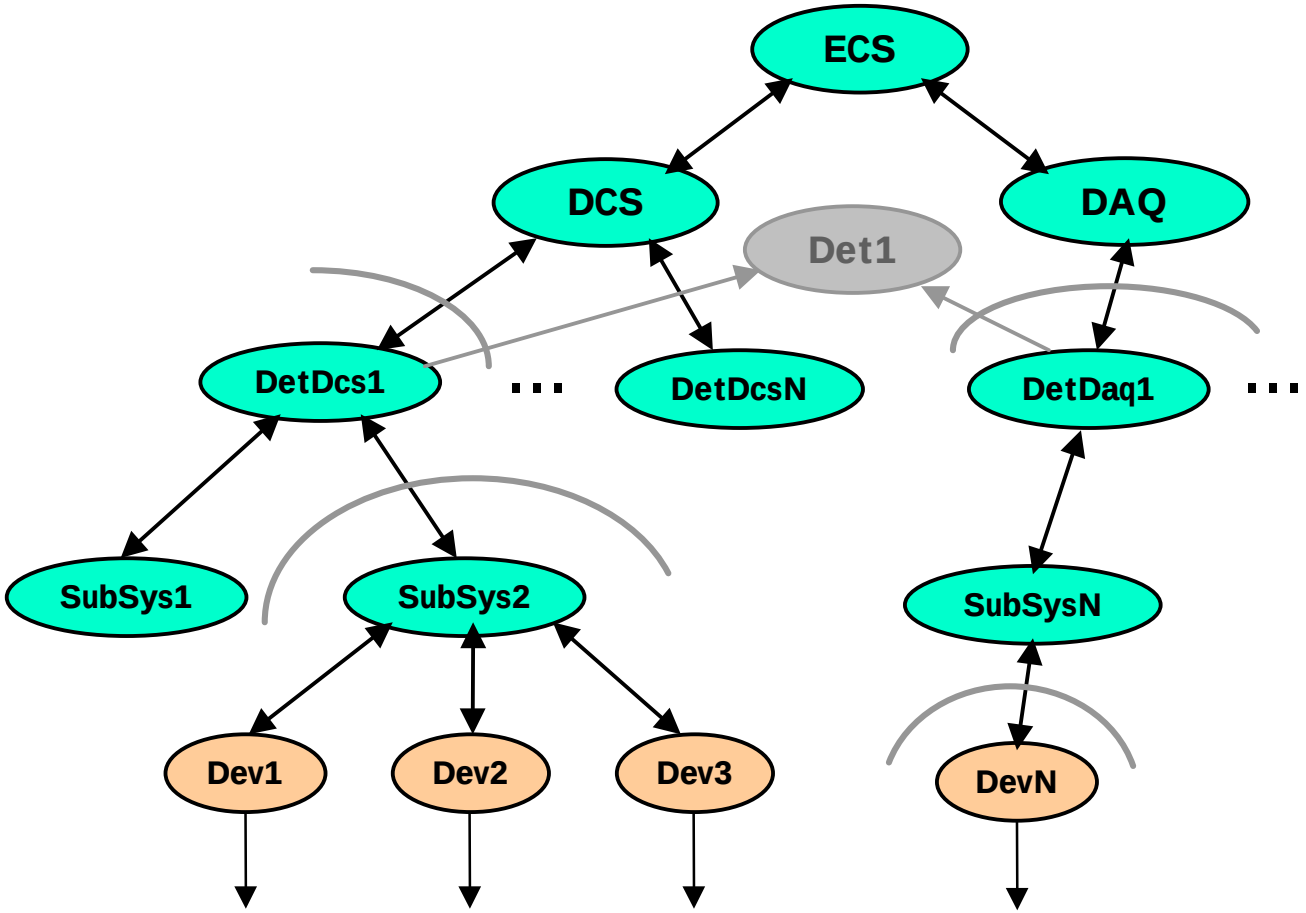
Control Framework (cont.)

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



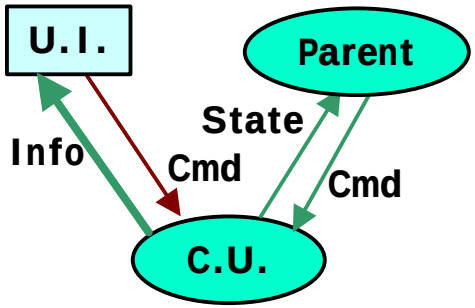
Clara Gaspar, May 2000

Control Unit Operation Modes

- **Control Units:**
  - Publish State and Information
  - Can receive commands

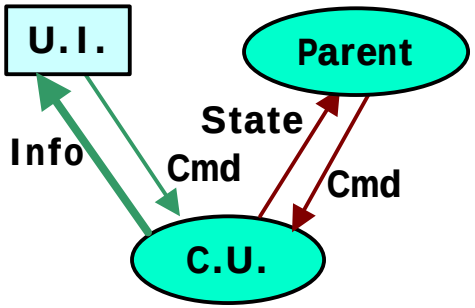
Normal Operation

- Hierarchical control only



Partitioned

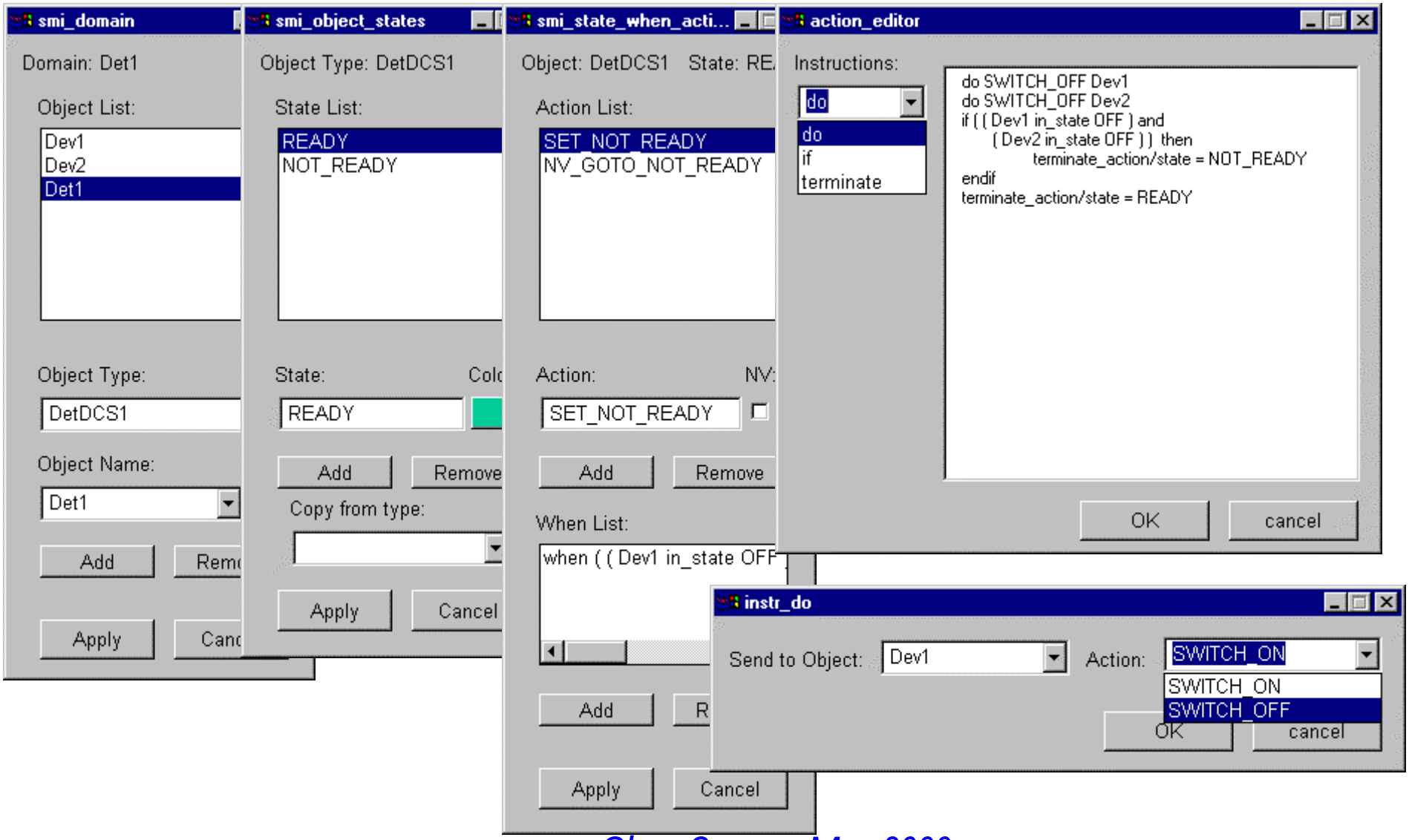
- No Hierarchical control
- Control from a “local” U.I.



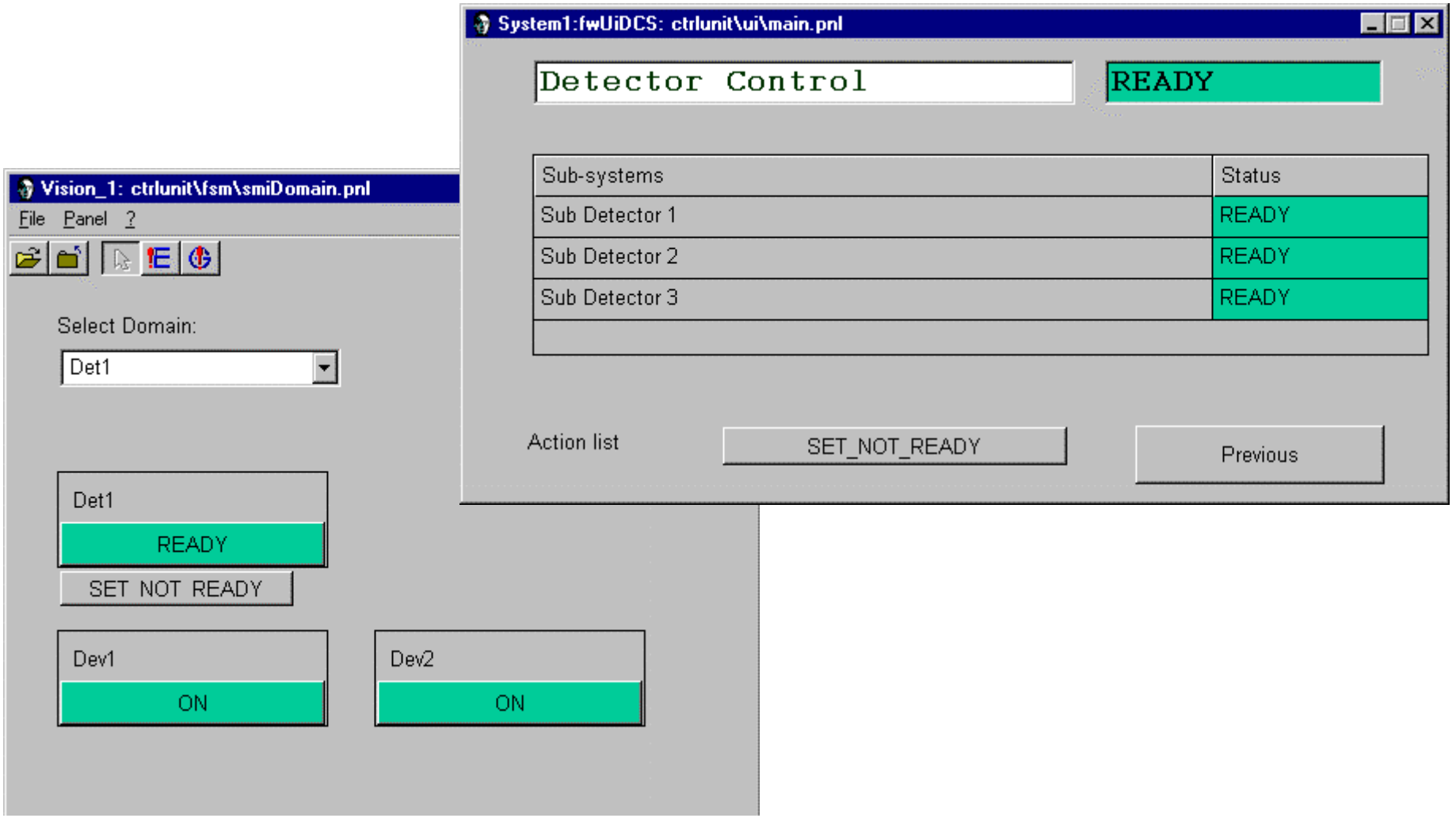
Control Unit tasks

- 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

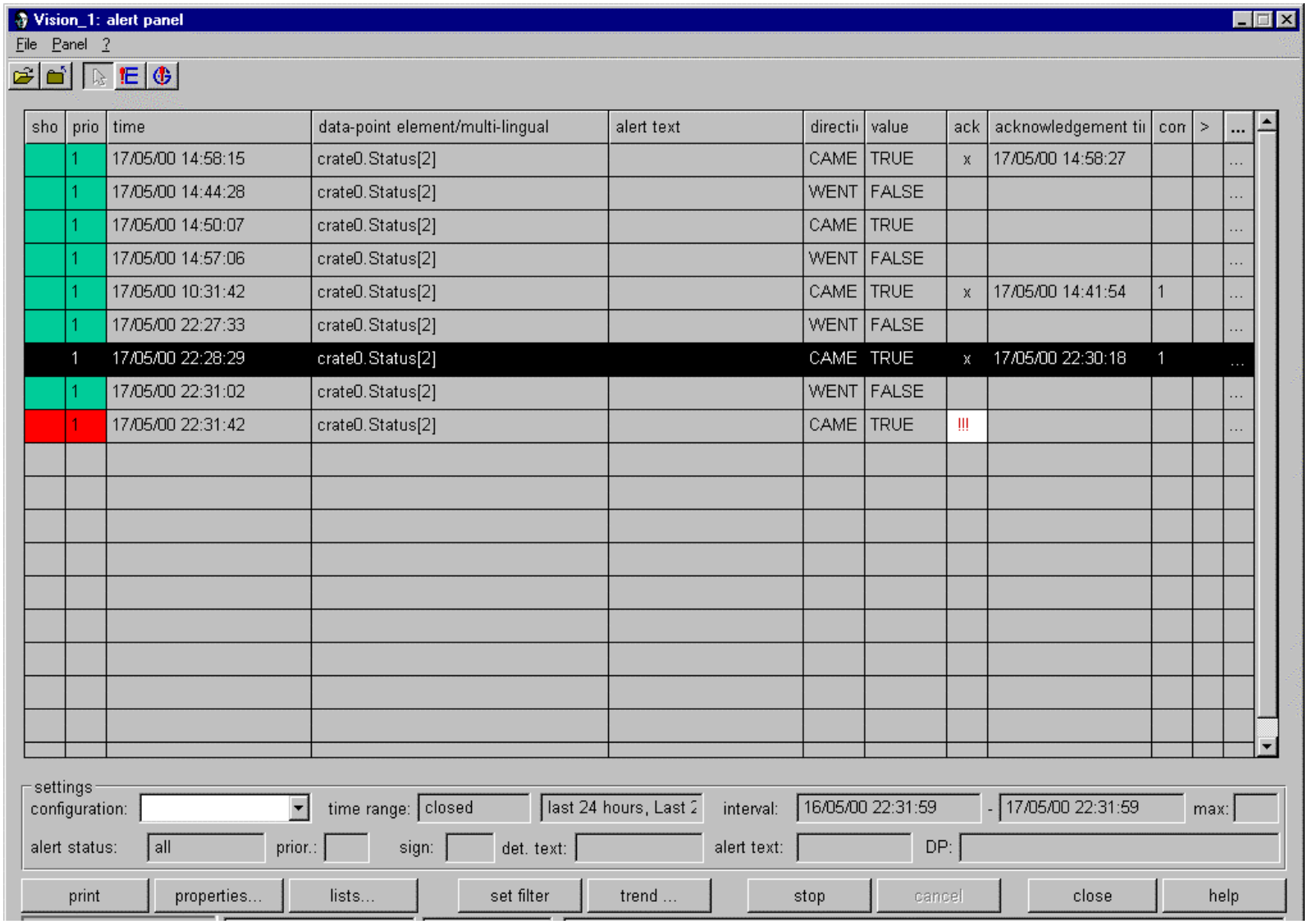
Modeling


smi_domain
Domain: Det1
Object List:
Dev1
Dev2
Det1
Object Type:
DetDCS1
Object Name:
Det1
Add
Apply
smi_object_states
Object Type: DetDCS1
State List:
READY
NOT_READY
State:
READY
Add
Remove
Copy from type:
Apply
Cancel
smi_state_when_acti...
Object: DetDCS1 State: RE
Action List:
SET_NOT_READY
NV_GOTO_NOT_READY
Action:
SET_NOT_READY
Add
Remove
When List:
when ((Dev1 in_state OFF
Add
Apply
Cancel
action_editor
Instructions:
do
do
if
terminate
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
OK
cancel
instr_do
Send to Object:
Dev1
Action:
SWITCH_ON
SWITCH_ON
SWITCH_OFF
OK
cancel

Generated User Interfaces



Alarm Display



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