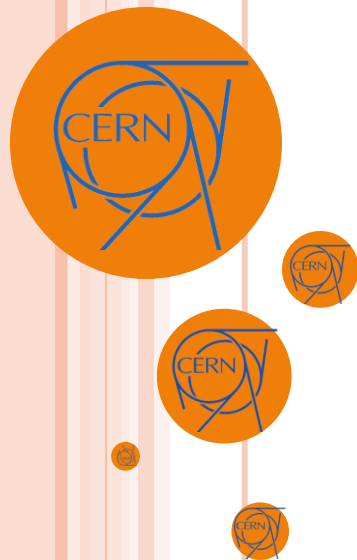


# UNICOS: UNIFIED INDUSTRIAL CONTROL SYSTEM CPC (CONTINUOUS PROCESS CONTROL)

**BASIC COURSE  
SESSION 0:**

**UCPC FROM SPECS TO IMPLEMENTATION**



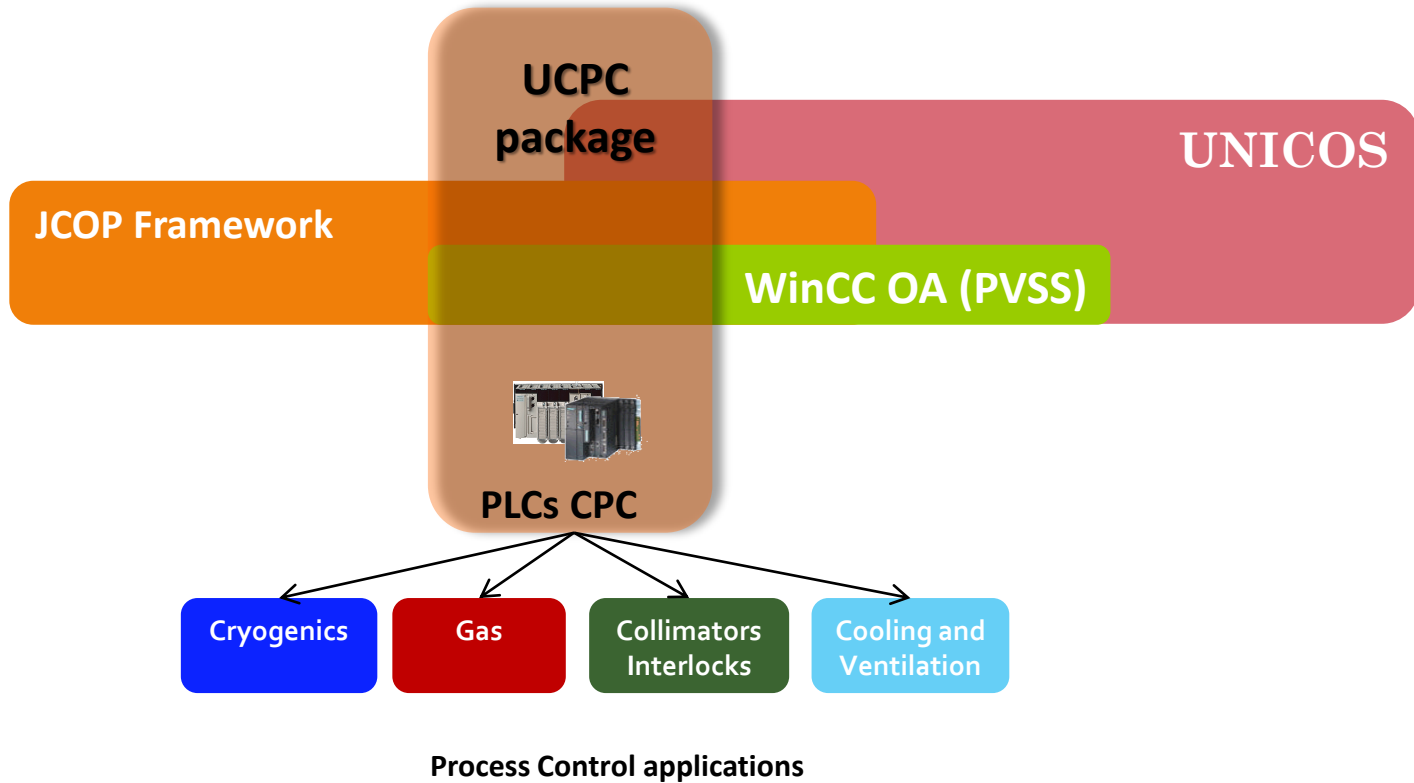
**UCPC 6**

UNICOS-Continuous Process Control

*CERN, UNICOS team*

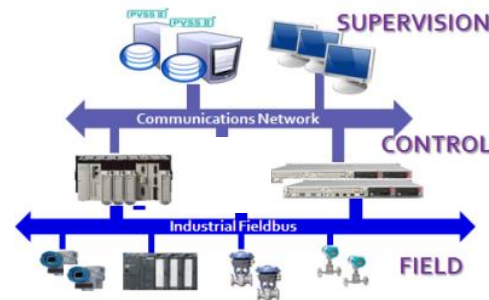
- **Process.** Methods of changing or refining raw materials to create end products.
- Process control refers to the methods that are used to control process variables when manufacturing a product.
- Manufacturers control the production process for several reasons:
  - Reduces variability: increase **quality**
  - Increases **efficiency**: increase profits
  - Ensures **safety**: optimize conditions

- **UNICOS** is a framework to create control applications.
- **UNICOS-CPC (Continuous Process Control)** is basic package to develop integrated PLC based process control applications.



# A LOOK TO THE PAST

- [1998] **UNICOS** (**UN**ified **I**ndustrial **C**ontrol **S**ystem) was born at CERN as a need to develop the LHC cryogenics control system. The **goal** was to create an industrial control system covering the three layers of the typical automation pyramid.



- [2002] UNICOS replace the PcVue SCADA and adopt ETM's PVSS under the CERN recommendation.
- [2004] UNICOS offered the choice of creating applications based on SIEMENS S7 PLCs
- [2009] UNICOS turned into *de facto* **standard framework** to develop industrial control applications at CERN:  
Cryogenics, Cooling, HVAC, Vacuum, Interlocks,...
- [2010] The UNICOS framework has been extended to other kinds of applications (i.e.: supervisory: [QPS,SURVEY], monitoring, ...)
- [2012] **Re-engineering** process

## Supervision Layer

WinCC OA SCADA

## Control Layer

Siemens S7-300, S7-400, S7-1500

Schneider Premium, Quantum, M580

Codesys (Somachine, TwinCat3)

## Local operation

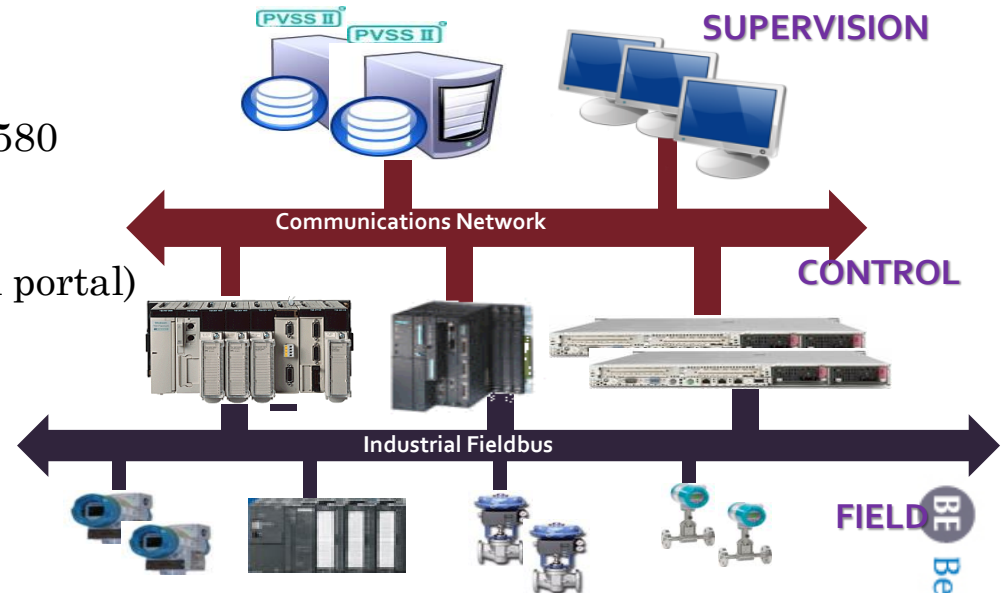
Simatic HMI (WinCC flexible, TIA portal)

Schneider Magelis

## Industrial communications

Profibus, Profinet, Ethernet/IP

Modbus TCP

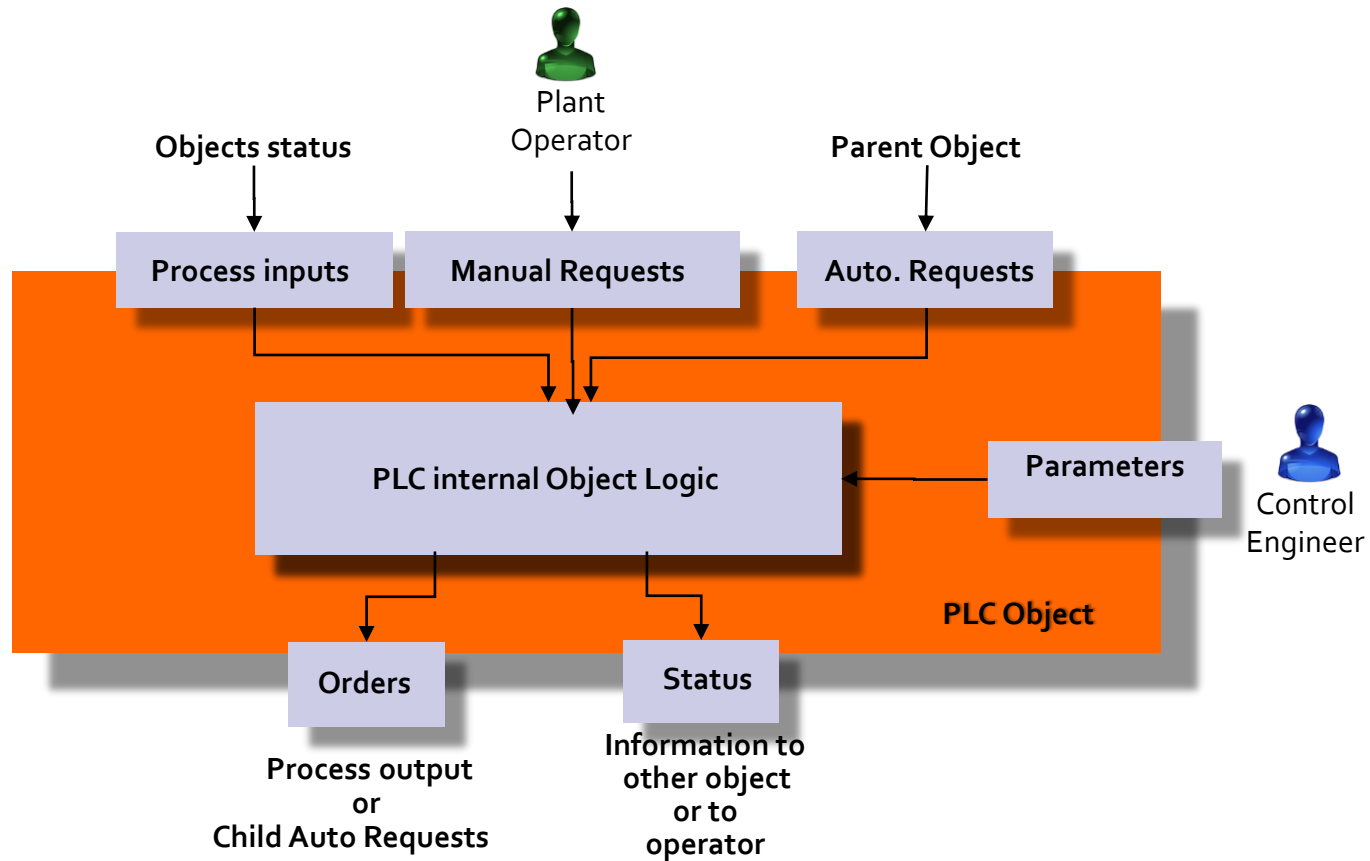


- LHC Cryogenics
- Detector and Test facilities cryogenics
- Magnet Control System
- Vacuum installations : ATLAS, CMS, ISOLDE
- LHC collimators: Environmental temperatures
- ATLAS Big wheels (motion)
- AMS servomotors control
- Detector gas control systems
- Cooling and HVAC installations
- Winding machines: HTS cable (hybrid with a Safety system)
- SM18 Magnet test bench protection system.

- UNICOS CPC provides libraries (control and supervision layers)
- A well defined set of **standard device types** (objects), modeling most of the **equipment** and needs of continuous processes and the **relationships** between them.
  - **I/O Objects**
    - ✓ Digital I/O
    - ✓ Analog I/O
  - **Field Objects**
    - ✓ OnOff
    - ✓ Analog
    - ✓ AnalogDigital
    - ✓ Local
    - ✓ AnaDO
  - **Control Objects**
    - ✓ Controller
    - ✓ Alarms
    - ✓ Process Control Object
  - **Interface Objects**
    - ✓ Parameter (Digital, Word, Analog)
    - ✓ Status (Word, Analog)
- A **formalized** way of :
  - Define the **control units** of a process (ISA-88 standard: Batch processes)
  - **Programming** the **specific process logic** for those units

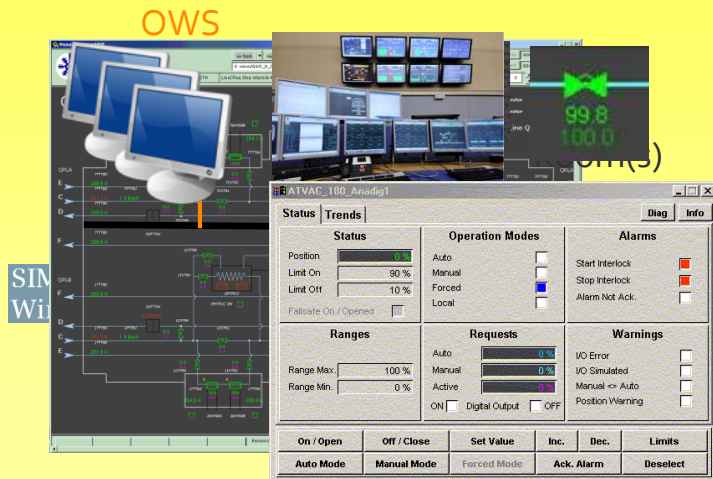


# UNICOS CPC OBJECT MODEL

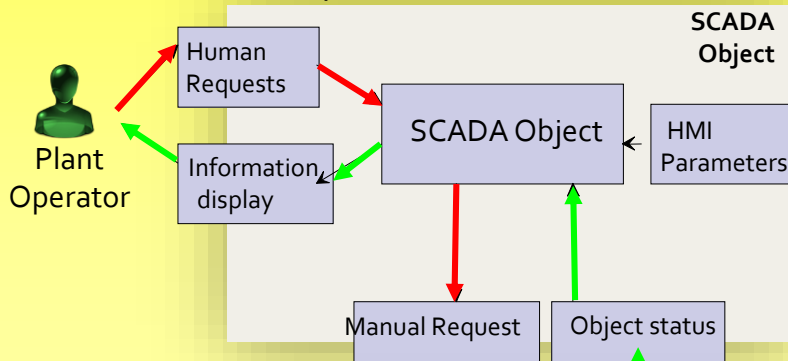


# OBJECTS & LAYERS INTEGRATION

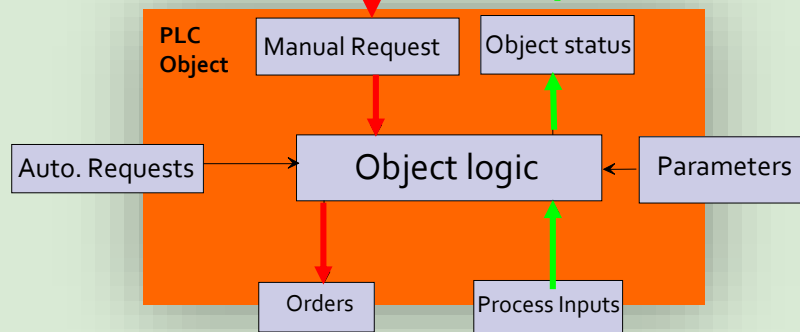
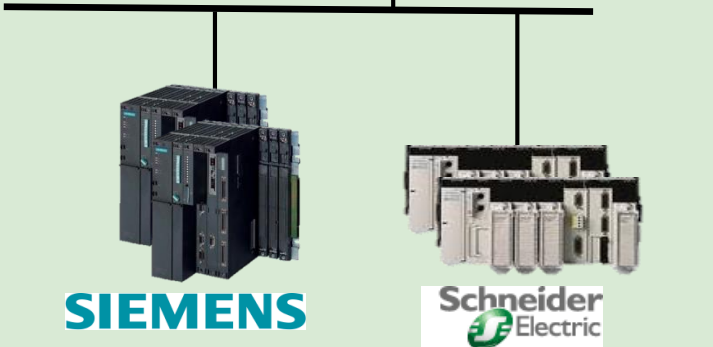
## Supervision Layer



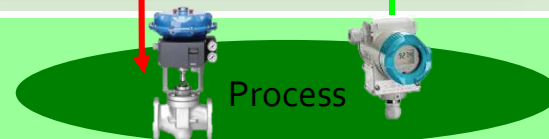
In the Supervision layer the object presents the relevant information to the operator and allow manual commands



## Control Layer



## Field Layer



- **I/O Objects**
- **Field Objects**
- **Control Objects**
- **Interface Objects**

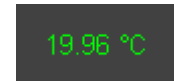
## ○ **Functionality**

- Base components
- PLC Periphery interface and/or internal memory variables

## ○ **Types**

- AI, AIR: Analog Input or Analog Input Real  
(*e.g. temperature transmitter*)
- DI: Digital Input  
(*e.g. end contact*)
- AO, AOR: Analog Output or Analog Output Real  
(*e.g. control valve position order*)
- DO: Digital Output  
(*e.g. onoff valve position order*)

*Widget examples*



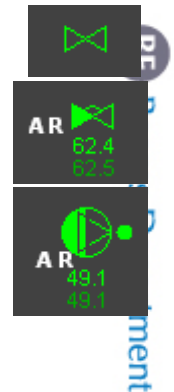
## ○ Functionality

- Model the real field equipments (e.g. pumps, valves...)
- As a general rule, the field objects are connected to the I/O Objects. No direct connection to the PLC periphery.

## ○ Types

- *OnOff*: Binary Objects  
(e.g. on/off valve, motor, pump)
- *Analog*: Analog objects  
(e.g. control valve, heater)
- *Anadig*: Analog inputs and Digital outputs objects  
(e.g. valves/heaters controlled by on/off pulses)
- *AnaDO* : Similar functionality of an OnOff + Analog object  
(Motor with VFD, Thyristor, Heater, etc.)
- *Local*: Field localized objects :  
(e.g. manual valve)

Widget examples



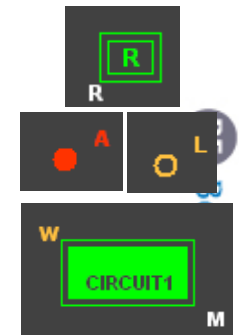
# CONTROL OBJECTS

- **Functionality**
  - Main objects holding the control logic
  - Feedback controllers
  - Handle the abnormal situations: Alarms and interlocks
  
- **Types**
  - *PCO*: Process Control Objects/Unit. It implements the control logic (e.g. *Compressor Station*)
  
  - *Controller*: feedback control objects (e.g. *PID controller*)
  
  - *AA, DA*: Analog/Digital Alarm Objects. It models alarms and interlocks. Analog alarms include alarm and warning thresholds (e.g. *Temperature Too High*)

## Options for an AA:

- Explicit threshold: Initialized in PLC and then modified from SCADA
- Logic: Set by control logic in the PLC
- APAR : Linked object APAR sets the value

Widget examples



## ○ **Functionality**

- Parameterization and status
- Can be connected to the periphery
- Light objects

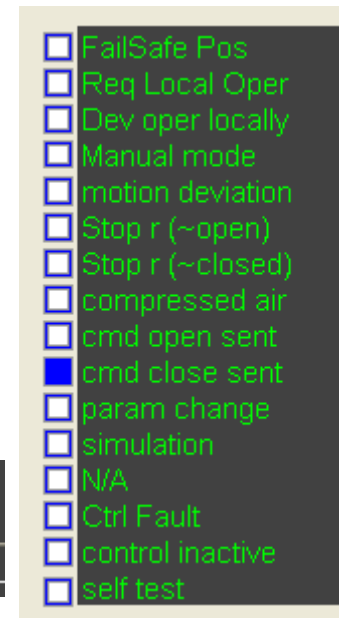
## ○ **Types**

- *DigitalParameter, WordParameter, AnalogParameter*: Parameters  
(e.g. *Threshold*)

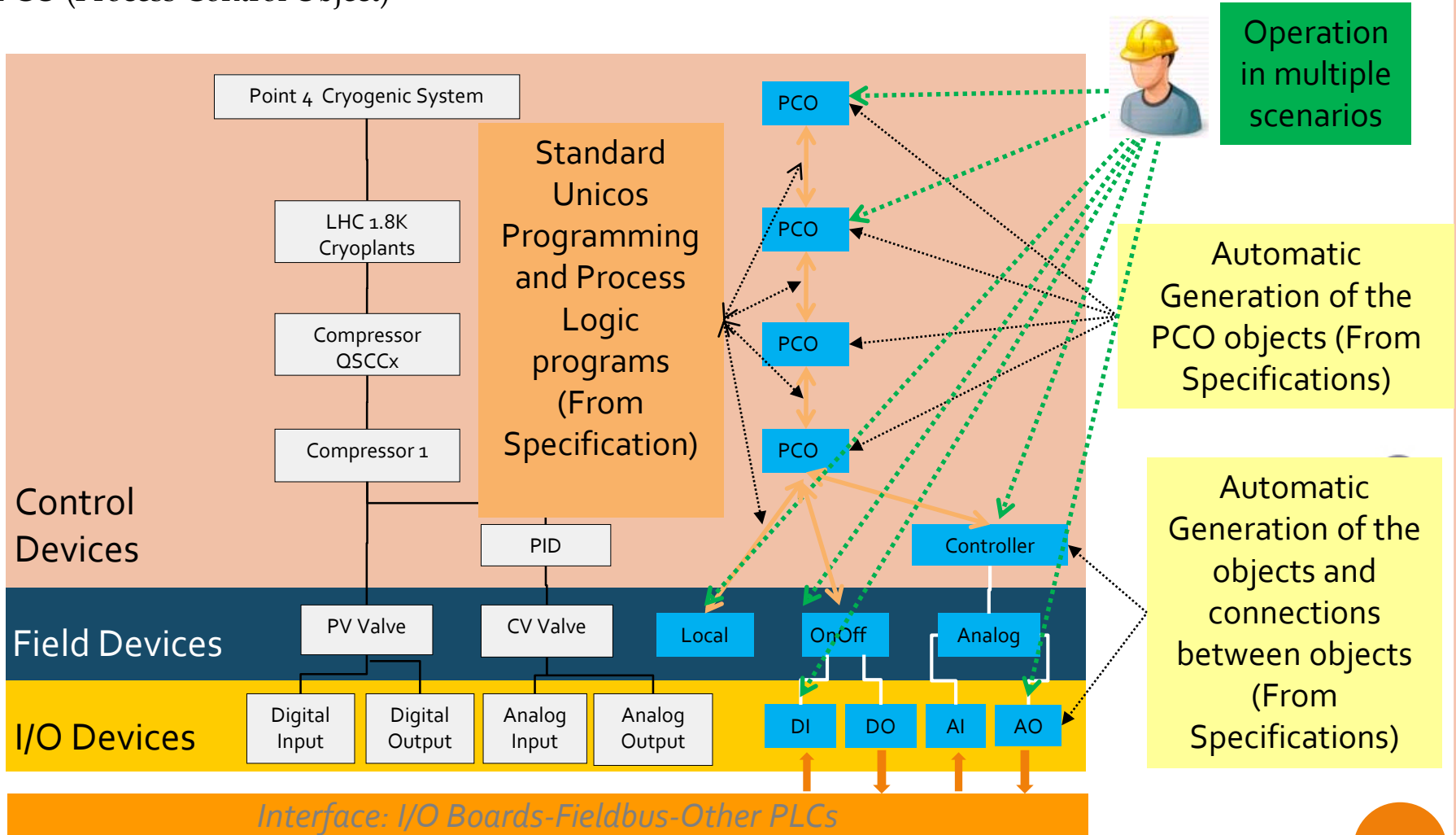
Can be set by an operator (SCADA -> PLC)

- *WordStatus, AnalogStatus*: Status  
(e.g. *stepper position, PA valve feedback*)  
PLC ->SCADA

*Widget examples*



- Each control module or equipment module is a device
- Equipment modules and Units are embedded in a unique object class: PCO (Process Control Object)

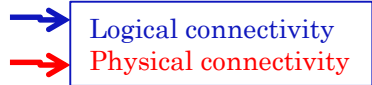




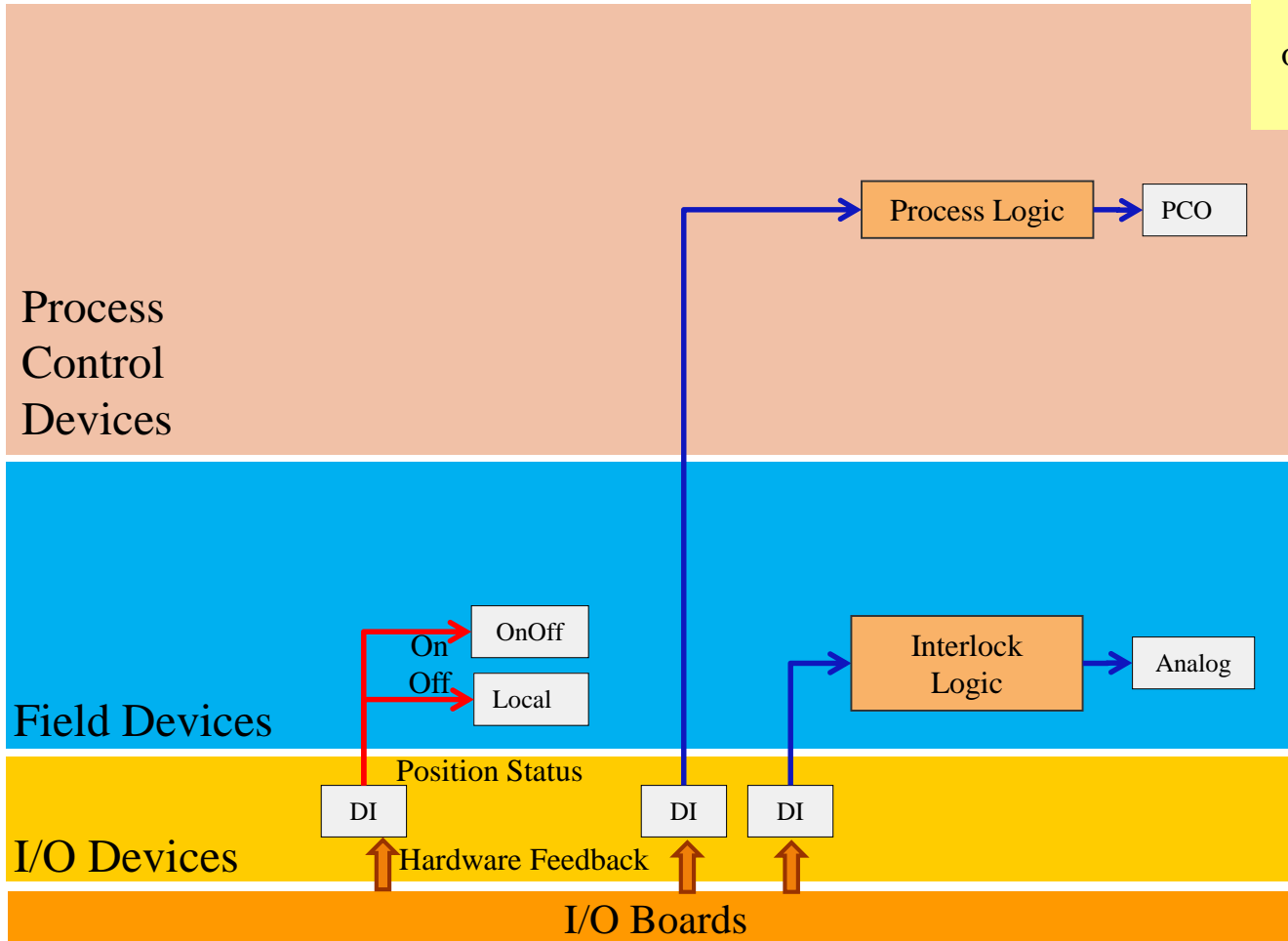
# OPERATION MODES

- **Auto Mode**
  - The object is driven by the control logic of a higher object of the hierarchy.
  - Interlocks apply to the request
- **Manual Mode** (requested by operators via the OWS)
  - The automatic return to the auto mode is possible by the control logic.
  - Interlocks apply to the request
- **Forced Mode** (requested by operators via the OWS)
  - The automatic return to the auto mode is impossible by the control logic.
  - Interlocks apply to the requests.
- **Local mode**
  - **Hardware Local Mode**
    - The object is driven locally by the process field (activated via a DI)
    - E.g. maintenance purposes
  - **Software Local Mode** (requested by operators via the Local panels)
    - The Local software is writing directly in the manual requests of the objects
    - **Priority** over "Auto" and "Manual" mode. The "forced mode" setup by the normal SCADA can override the software local mode.
    - Interlocks apply to the requests.

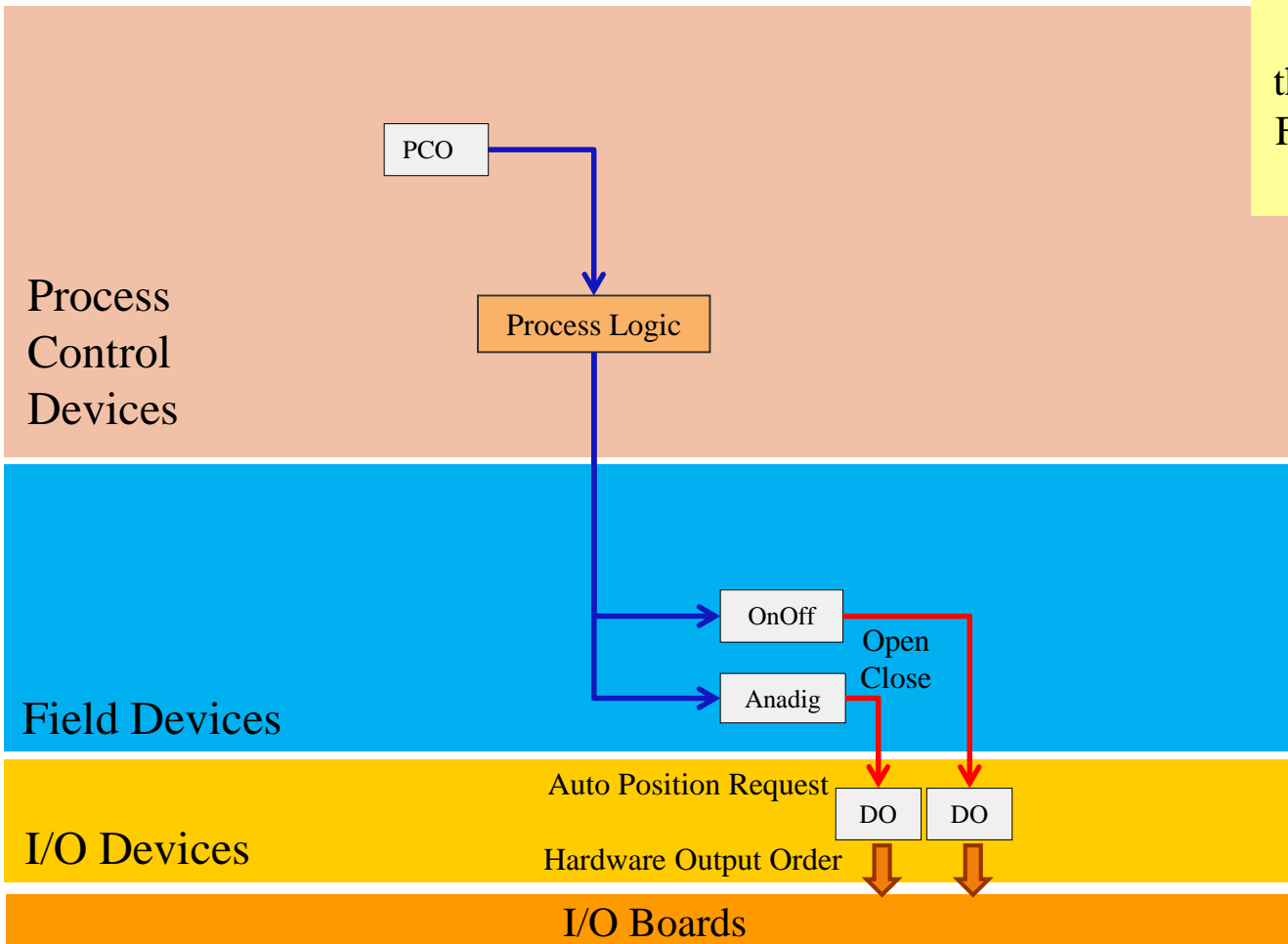
# DI OBJECT CONNECTIVITY



Status of DI Objects may be used by all objects anywhere in the program.



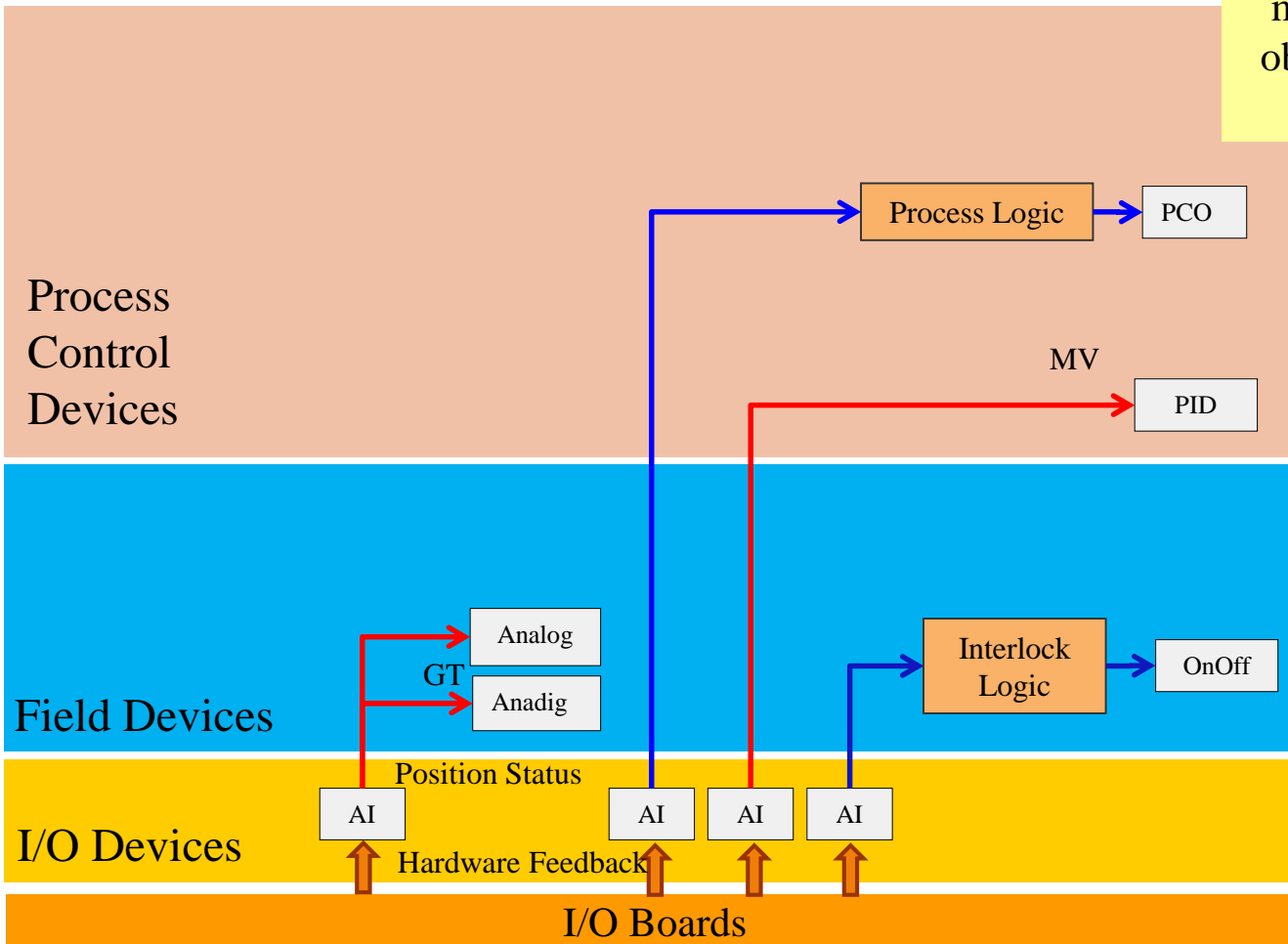
# DO OBJECT CONNECTIVITY

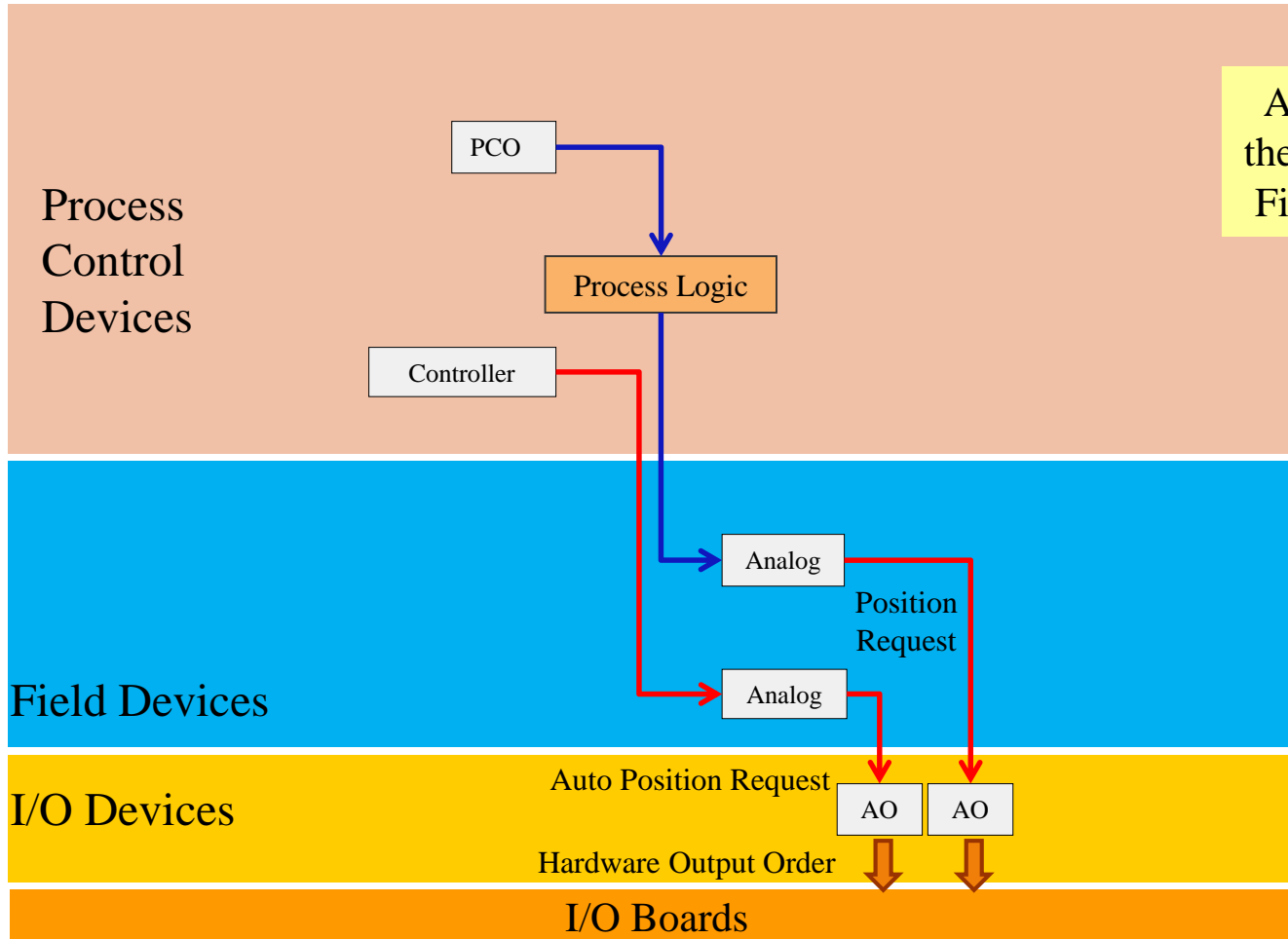


DO Objects receive their orders only from Field Devices (OnOff & Anadig, AnaDO)

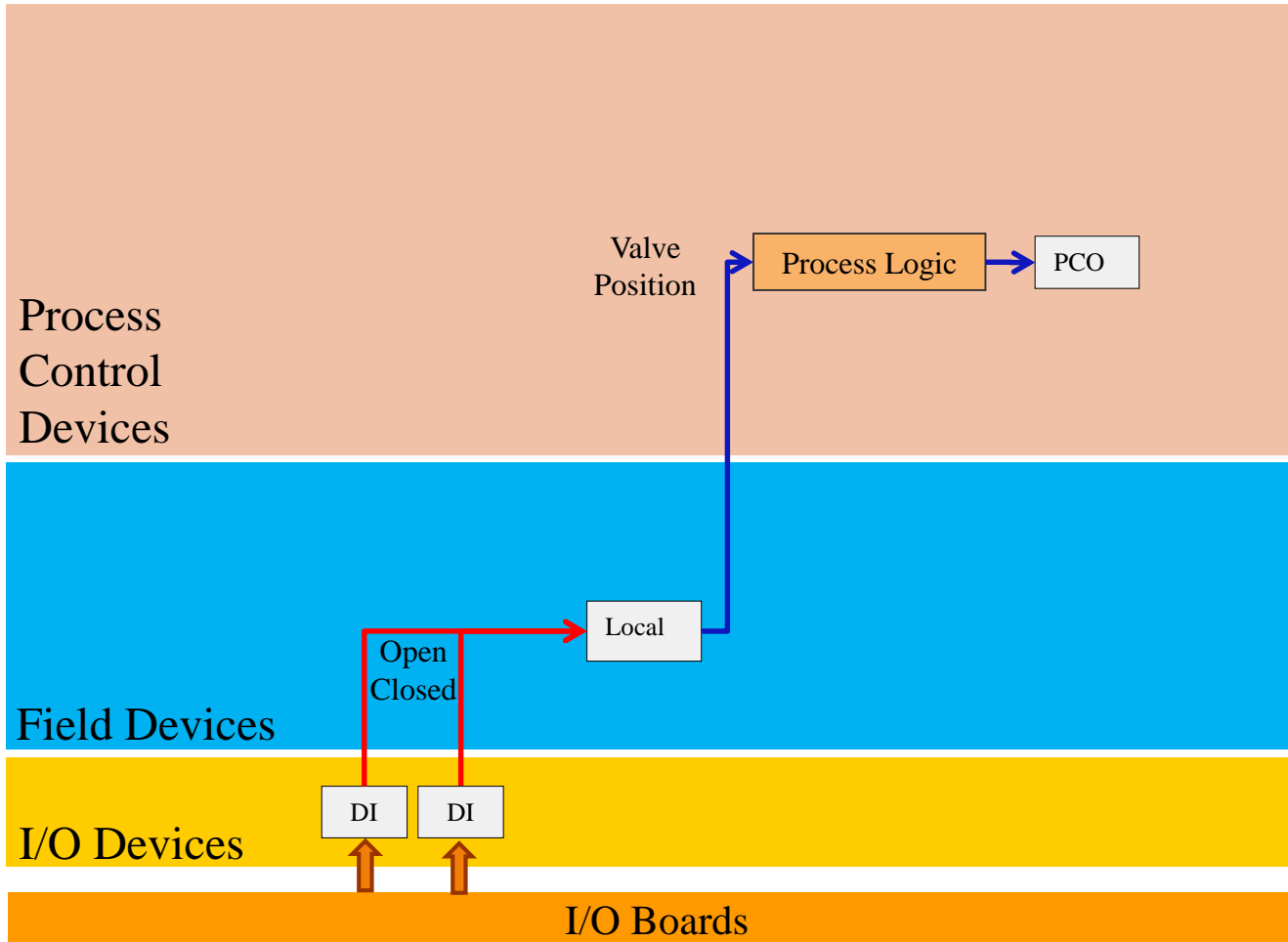
# AI OBJECT CONNECTIVITY

Status of AI Objects may be used by all objects anywhere in the program.



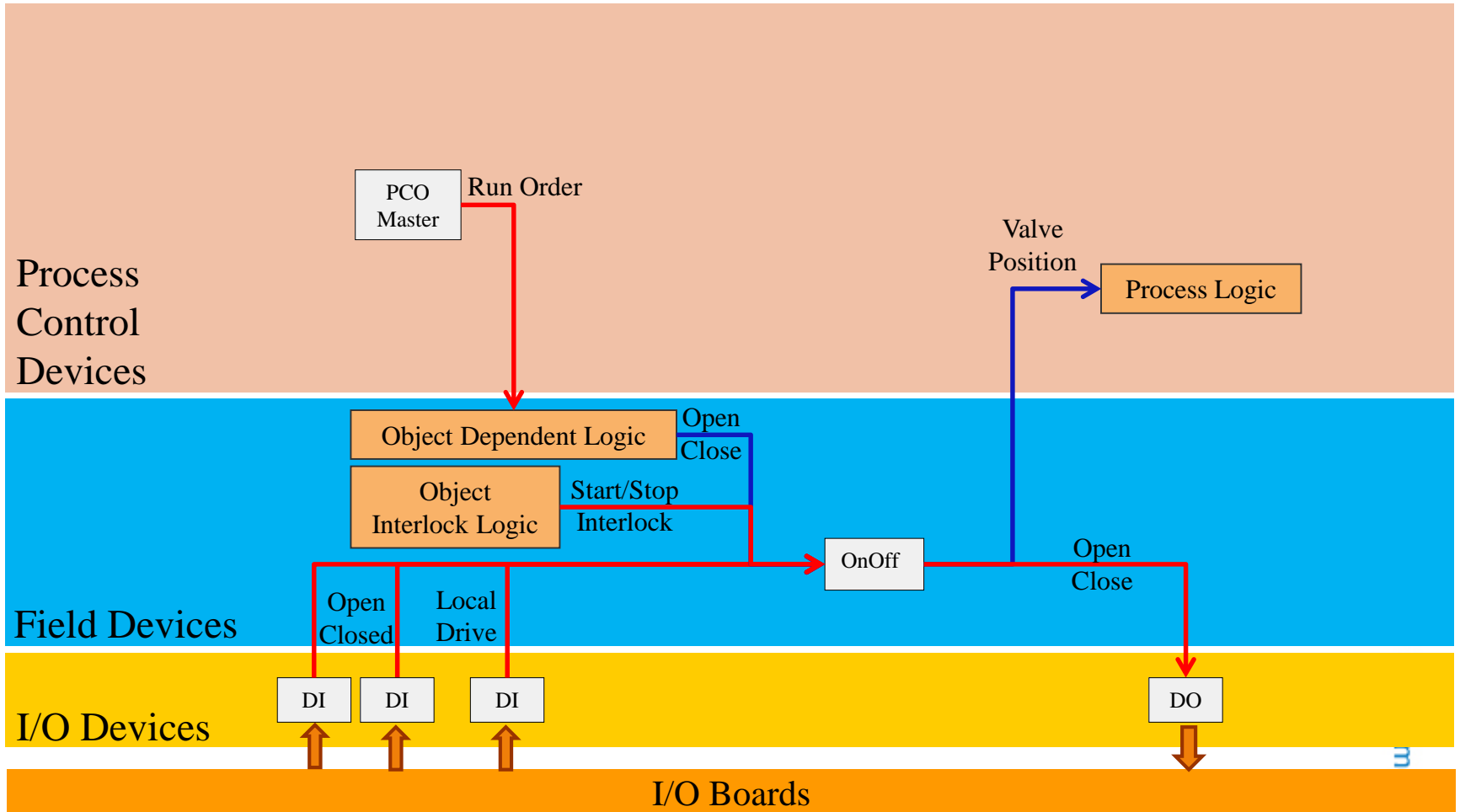


AO Objects receive their orders only from Field Device Analog

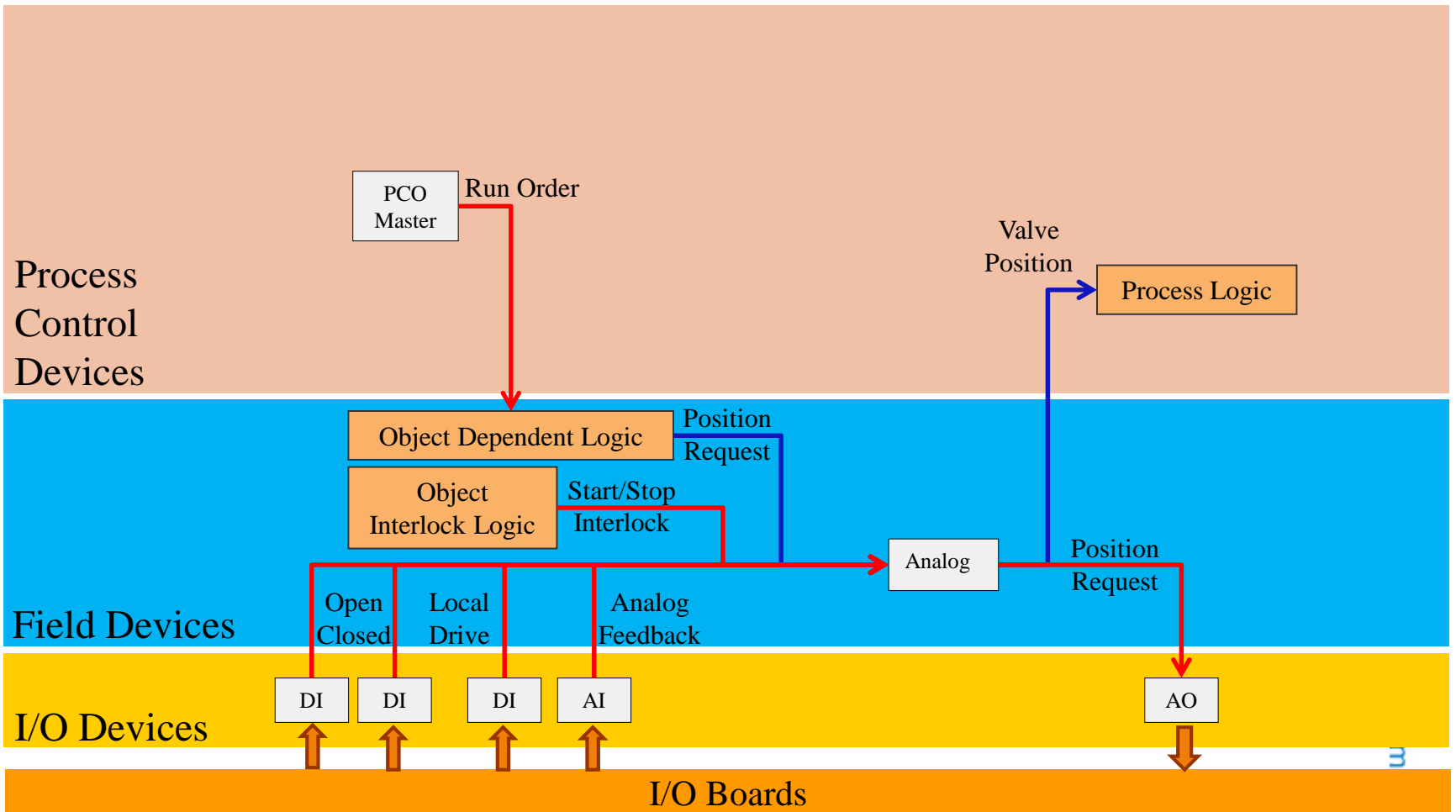


Status of Local Objects may be used by all objects anywhere in the program.

# ONOFF OBJECT CONNECTIVITY

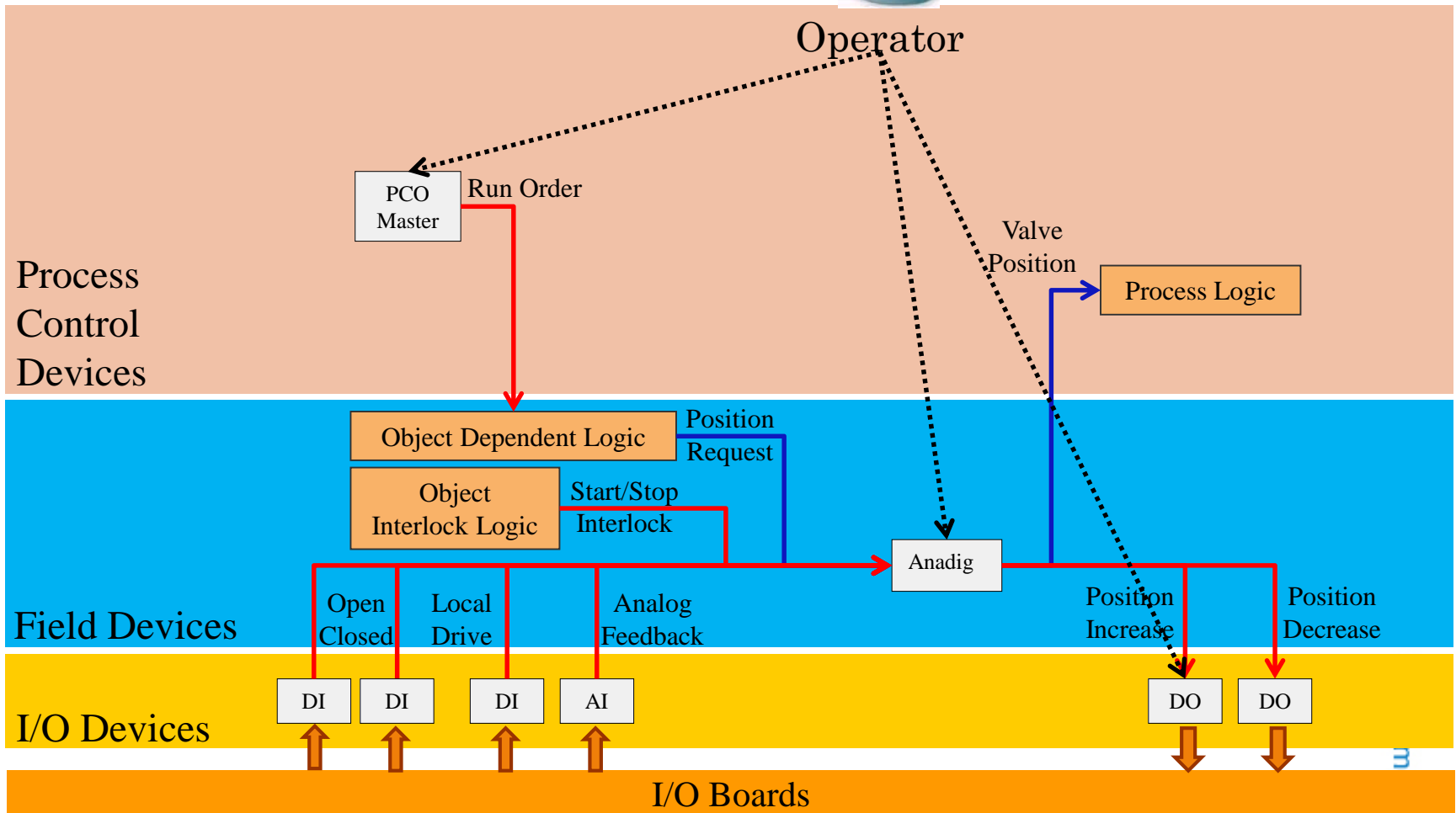


# ANALOG OBJECT CONNECTIVITY (NO PID)



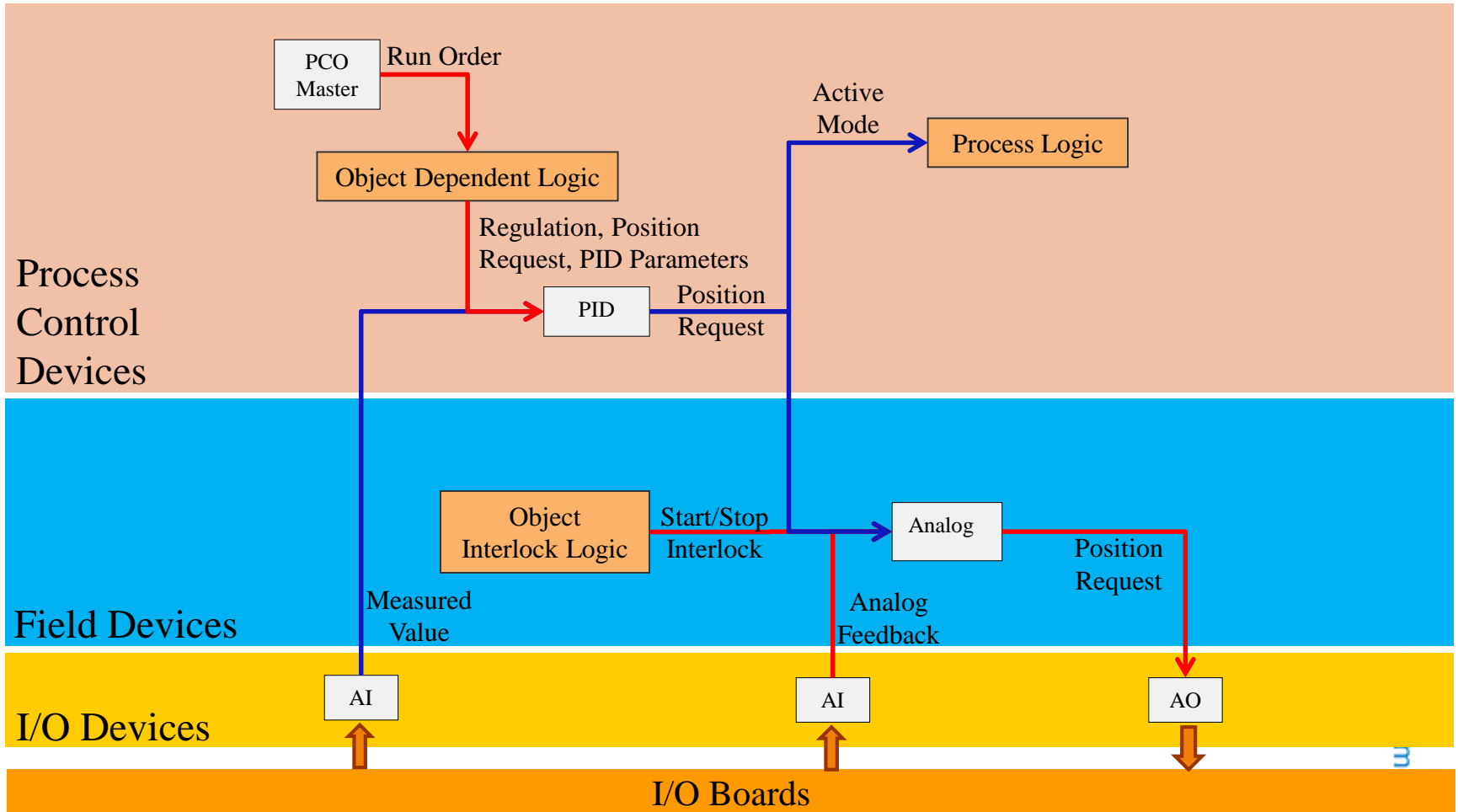


# ANADIG OBJECT CONNECTIVITY (NO PID)



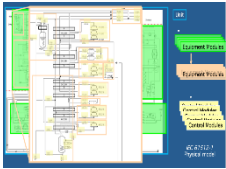
3

# CONTROLLER OBJECT CONNECTIVITY

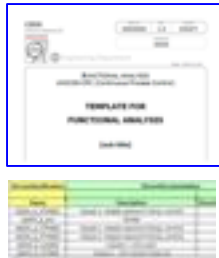


# UNICOS CPC ENGINEERING LIFE CYCLE

Decomposition



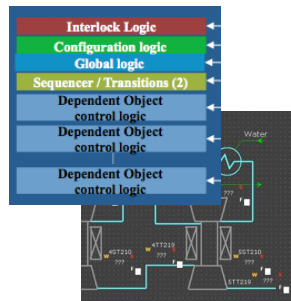
Specifications



Automatic Code  
Generation



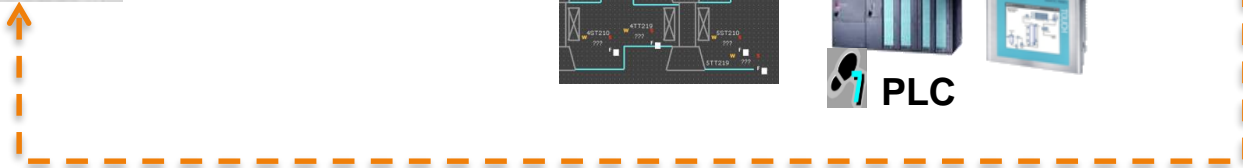
Logic specifics  
& Synoptics

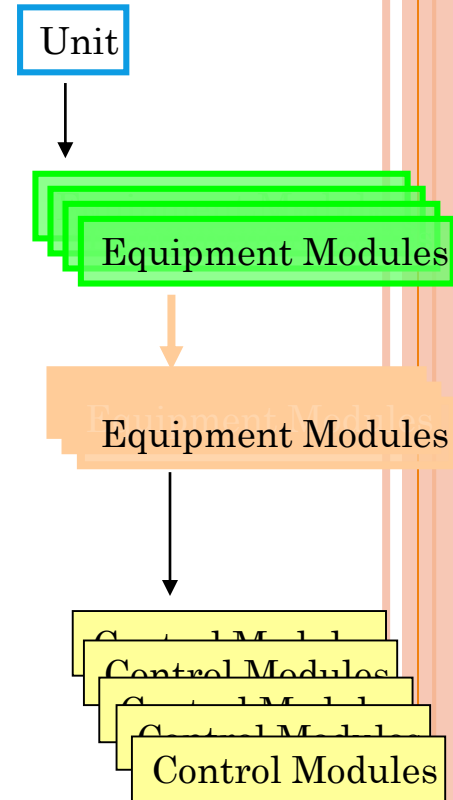
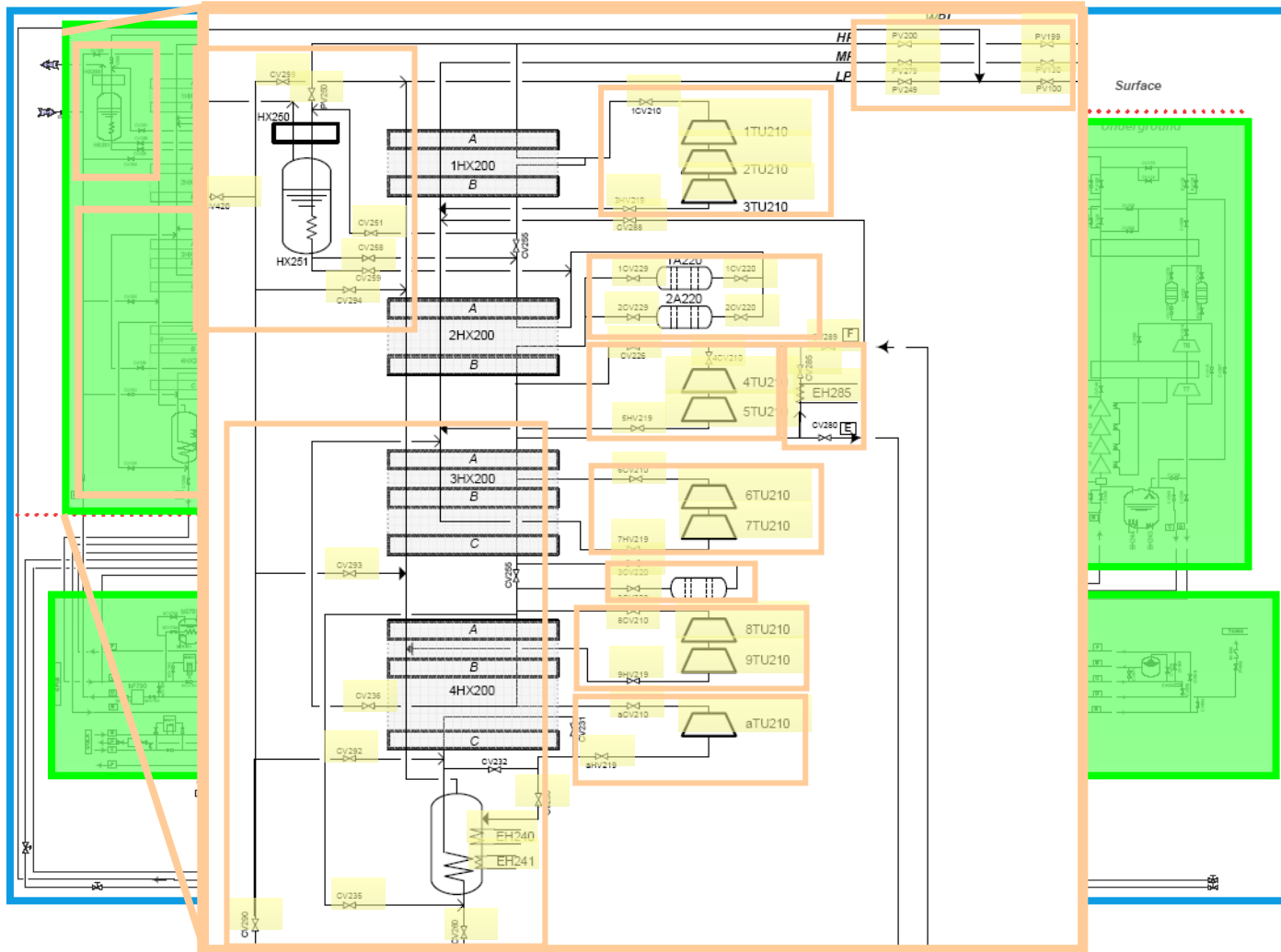


Deployment



Reverse  
Engineering





*IEC 61512-1*  
Physical model



# SPECIFICATIONS

DeviceIdentification	DeviceDocumentation			FEDeviceIOConfig		FEDeviceParameters					
	Name	Description	Electrical Diagram	Remarks	FE encoding type	InterfaceParam1	Range Min	Range Max	Raw Min	Raw Max	DeadBand (%)
QSDN_4_1TT4001	Vessel 1- Heater section1-Temp. control	AI1.0				%IW1.1.0	80	350	0	10000	0.025
QSDN_4_AI1	SPARE	AI1.1				%IW1.1.1	0	100	0	10000	0.025
QSDN_4_1TT4002	Vessel 1- Heater section2-Temp. control	AI1.2				%IW1.1.2	80	350	0	10000	0.025
QSDN_4_1TT4003	Vessel 1- Heater section3-Temp. control	AI1.3				%IW1.1.3	80	350	0	10000	0.025
QSDN_4_1LE400	Vessel 1- LN2 Level	AI1.4				%IW1.1.4	0	1350	0	10000	0.025
QSDN_4_1PT400	Vessel 1- LN2 Vessel Pressure	AI1.5				%IW1.1.5	0	4.0	0	10000	0.025

UNICOS CPC Specs (xls/xml file)

Functional Analysis + Logic specification  
(Word templates)

**CERN**  
CH1211 Geneva 23  
Switzerland

EDMS NO. **0000000** REV. **1.0** VALIDITY **DRAFT**

REFERENCE **XXXX**

Engineering Department

Date : 2010-11-24

**FUNCTIONAL ANALYSIS**  
UNICOS-CPC (Continuous Process Control)

**TEMPLATE FOR**  
**FUNCTIONAL ANALYSIS**

[sub title]

Engineering Department

2. PROCESS DESCRIPTION

2.4 Process decomposition

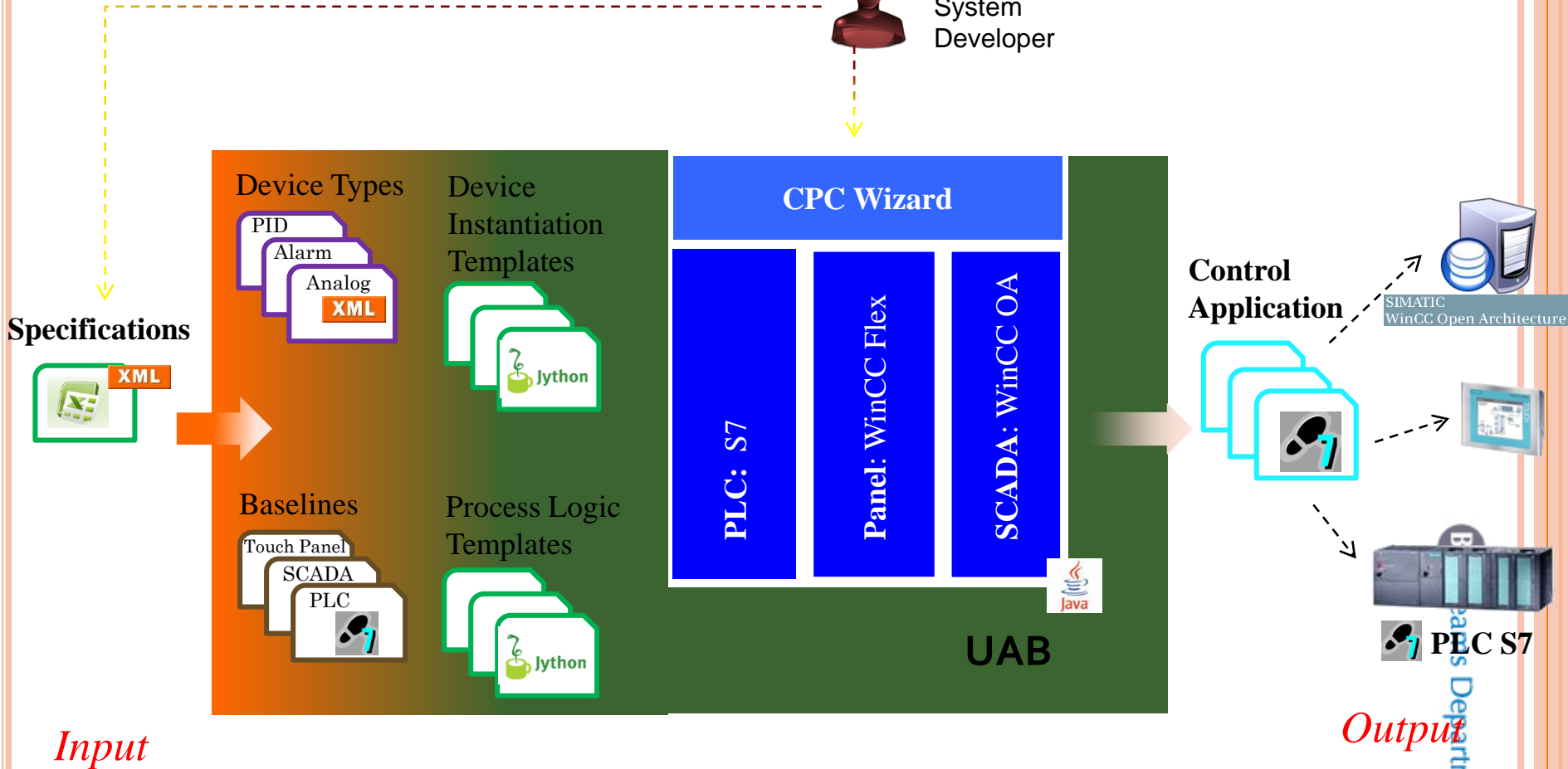
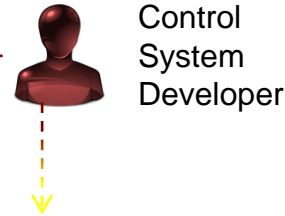
3.2 Operational States

3.2.4 Actuator operation

3.5 Unit Alarms

3.5.1 Unit hardware alarms

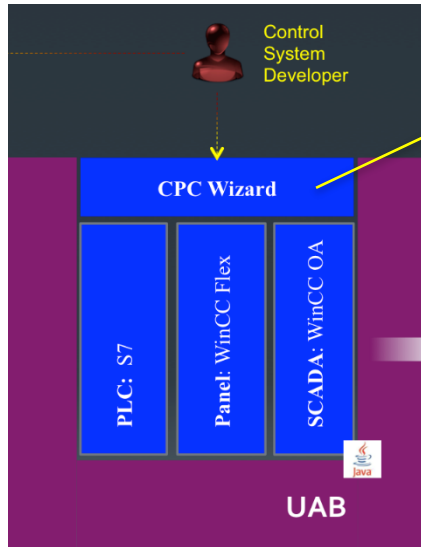
Name	Condition	Action*	Message
DNCT_FS1	ESSCOK Off	FS	equipment emergency s
DNCT_FS2	24VPwOn. Off	FS	Presence 24VDC Po
DNCT_FS3	24VIOOn	FS	Presence 24VDC I
DNCT_FS4	20Q6. Off	FS	Circuit breaker 24VDC for emerg
DNCT_FS5	26Q2. Off	FS	Circuit breaker 24VDC D



Input

Output

# WORKFLOW BASED ON WIZARDS



UAB CPC-Wizard v1.3.2-beta-02

CPC-Wizard: test - test v1.0

Unity Logic Generator  
Resources: 1.3.2-beta-02

General Data

Templates Folder: C:\temp\wizard\Release\Schneider\Resources\UnityLogicGenerator\ Open

User Templates Folder: C:\temp\wizard\Release\Schneider\Resources\UnityLogicGenerator\ Open

Output Folder: C:\temp\wizard\Release\Schneider\Output\UnityLogicGenerator Open

Output File: C:\temp\wizard\Release\Schneider\Output\UnityLogicGenerator\plc\_ Open

Process Semantic Rules:  Generation Language: ST

Import and Generate

Master	Section	Type	Master	Logic File
DEMON_1_DemonPCO	DEMON_1_Demon...	Interlock Logic	DEMON_1_Demon...	SchLogic_IL_Stand...
DEMON_1_PCO3	DEMON_1_Demon...	Configuration Logic	DEMON_1_Demon...	SchLogic_CL_Stan...
DEMON_1_PCO1	DEMON_1_Demon...	Basic Logic	DEMON_1_Demon...	SchLogic_BL_Stan...
DEMON_1_PCO2	DEMON_1_Demon...	Instantiation	DEMON_1_Demon...	SchLogic_INST_St...
	DEMON_1_Demon...	Global Logic	DEMON_1_Demon...	SchLogic_GL_Stan...
	DEMON_1_Demon...	Transition Logic	DEMON_1_Demon...	SchLogic_TL_Stan...
	DEMON_1_Demon...	Sequencer Logic	DEMON_1_Demon...	SchLogic_SL_Stan...
	DEMON_1_Demon...	Common Depend...	DEMON_1_Demon...	SchLogic_CDOL_St...
	DEMON_1_A1_DL	Analog	DEMON_1_Demon...	SchLogic_Analog_...
	DEMON_1_A5_DL	Analog	DEMON_1_Demon...	SchLogic_Analog_...
	DEMON_1_AD1_DL	AnalogDigital	DEMON_1_Demon...	SchLogic_AnalogDi...
	DEMON_1_Ctrl1_DL	Controller	DEMON_1_Demon...	SchLogic_Controlle...
	DEMON_1_PCO3_DL	ProcessControlObject	DEMON_1_Demon...	SchLogic_ProcessC...
	DEMON_1_OO4_DL	OnOff	DEMON_1_Demon...	SchLogic_OnOff_S...

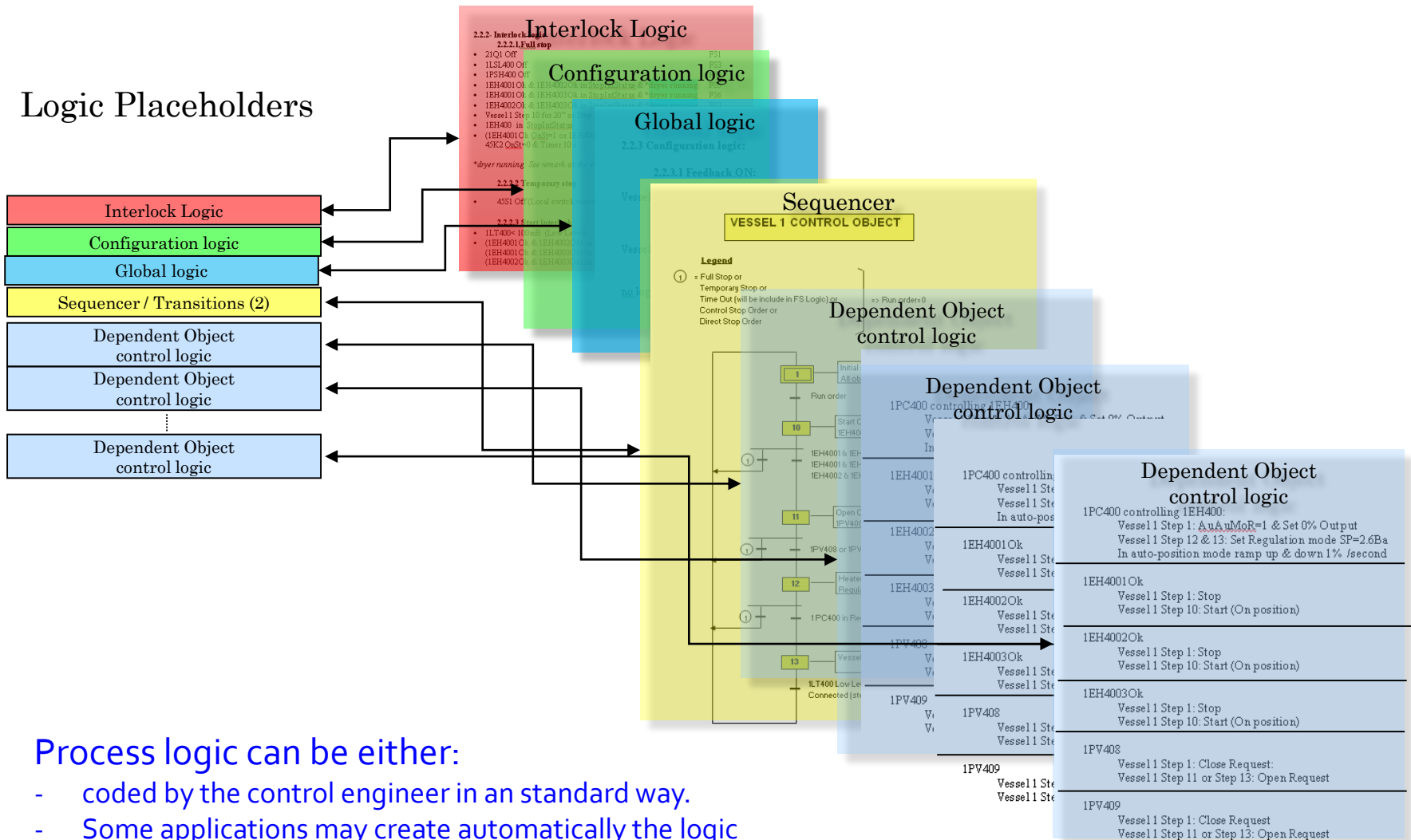
Filter: Interlock Logic, Configuration Logic, Basic Logic, Instantiation, Global Logic, Transition Logic

Generation Status

Instance Generator, Logic Generator, WinCC OA Gener..., WinCC Flex Gene...

Back, Generate, Exit

## Logic Placeholders



## Process logic can be either:

- coded by the control engineer in an standard way.
- Some applications may create automatically the logic based on **templates**. These templates are based on Python





# HMI SYNOPTICS

- Manual intervention (or automatic if known a priori)

TOUCH

Synoptic design

- by drag & drop (manual operation)
- Automatically created (xml)

# MORE INFORMATION

- Check out the web page:
  - <http://www.cern.ch/unicos>



## About UNICOS

### Introduction

**UNICOS (UNified Industrial Control System)** is a CERN-made framework to develop industrial control applications. It deals with the two upper layers of a classical control system: Supervision and Control. UNICOS proposes a method to design and develop the control application which will run in commercial off-the-shelf products (e.g. SCADA and PLCs). The framework employs terminology and models of the ISA-88 standard for batch control systems.

The goal of UNICOS is to standardize the development of control applications at CERN by:

- Emphasize good practices for both, design and operation, of the continuous process control applications
- Reduce the cost of automating continuous processes (e.g. cooling, HVAC...)
- Optimize life-cycle engineering efforts (e.g. using automatic code generation tools)