UNICOS: UNIFIED INDUSTRIAL COCONTROL SYSTEM

CPC (CONTINUOUS PROCESS CONTROL)

BASIC COURSE

SESSION 0:

UCPC FROM SPECS TO IMPLEMENTATION

UNICOS-Continuous Process Control

CERN EN/ICE group

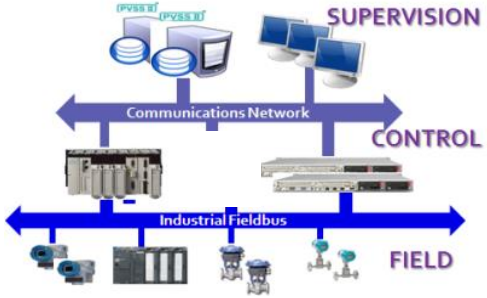
Outline

- **UNICOS vs. UNICOS-CPC**
- Objects main functionality and connectivity
- From specs to implementation: Overview
  - Life cycle
  - Generation Tools

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.

PcVue
SIEMENS
Schneider Electric



• [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

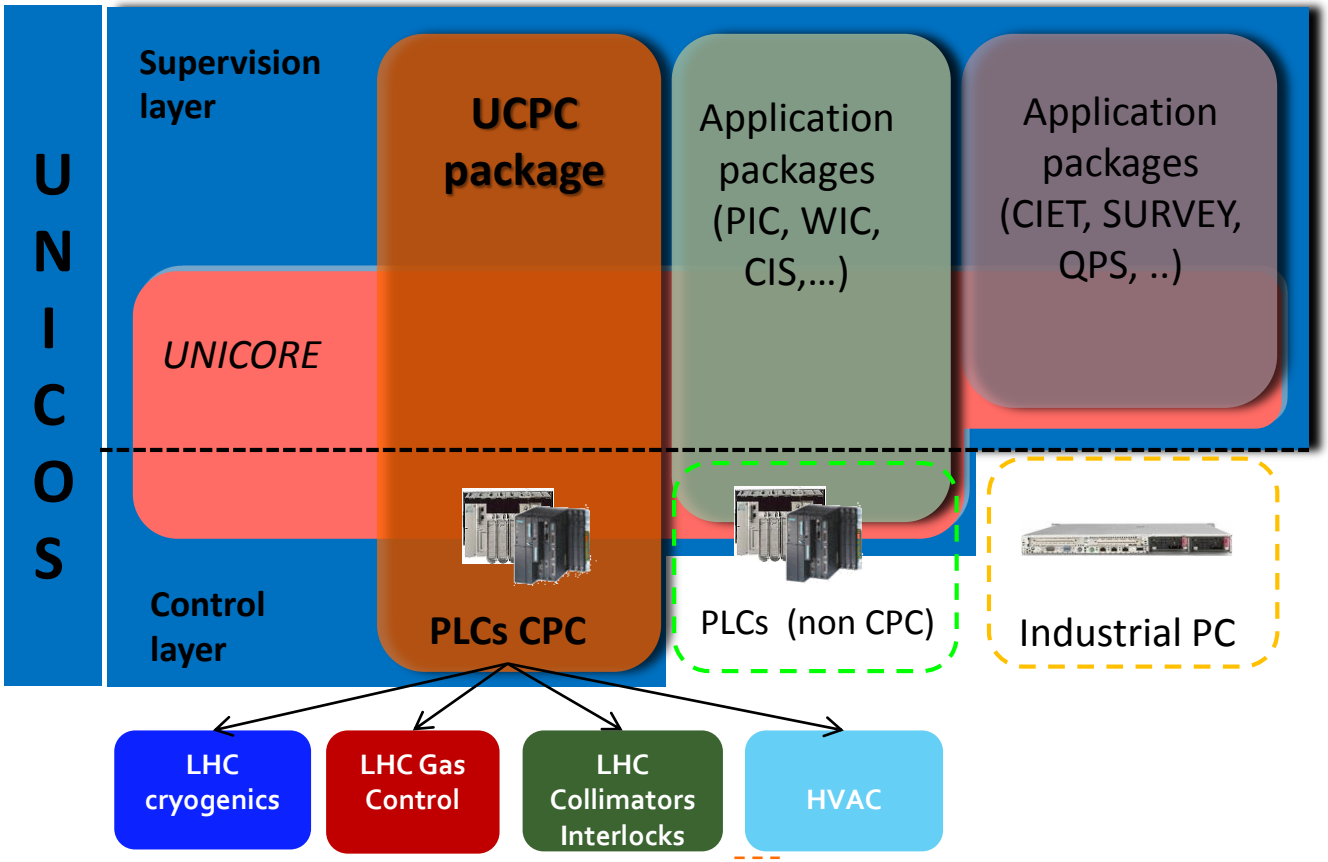
A FLAVOR OF APPLICATIONS

- Many UNICOS CPC applications done:
  - 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)
  - ….

UNICOS AND UCPC

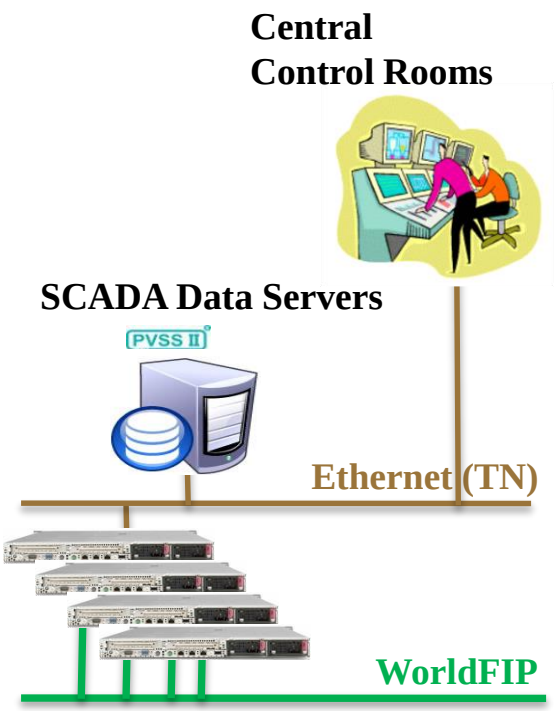
- UNICOS is a framework to create control applications

  UCPC: A basic package (**Continuous Process Control**) to develop integrated process control applications.

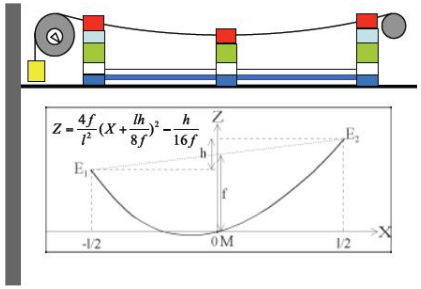


USE CASE: SURVEY (NO PLC)

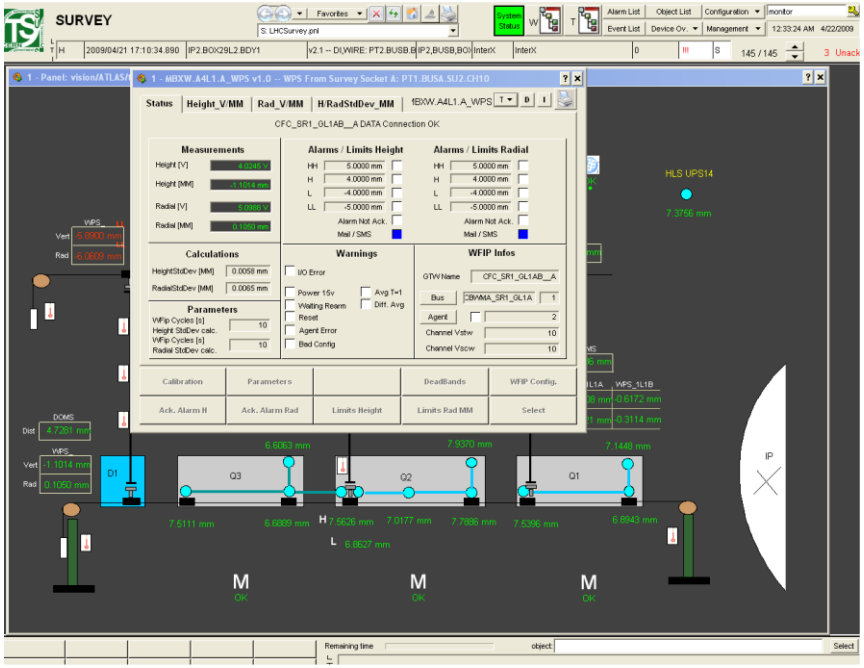
- LHC Inner Triplet magnet alignment (tolerance ~ 100 micron)
- Front End: FESA devices (DOMS, WPS, HLS, Steiner, ZTS, ESTOP, …)



General Architecture



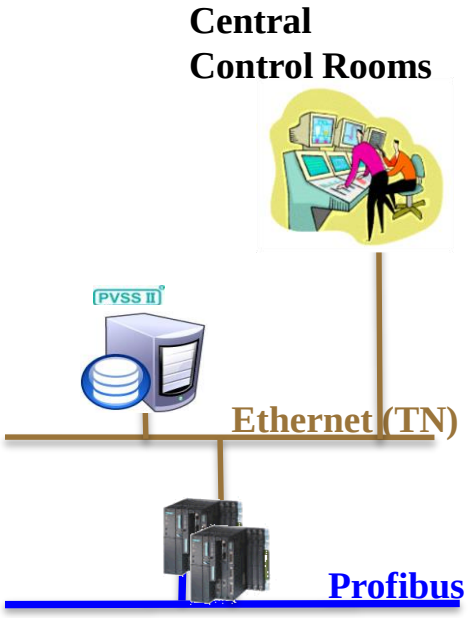
WPS: streched-wire pos. system

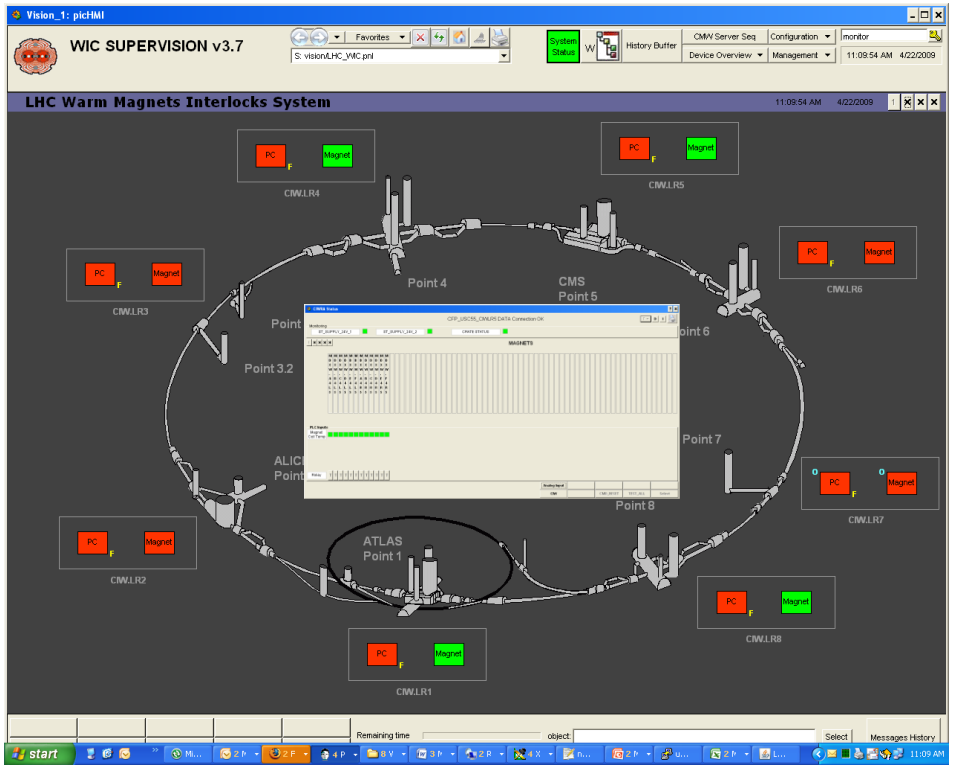


USE CASE: WIC (PLC NON UNICOS)

- Warm Interlock controller: monitors the magnet and safeguards it from overheating
- **PLC** is **not UNICOS-like**.

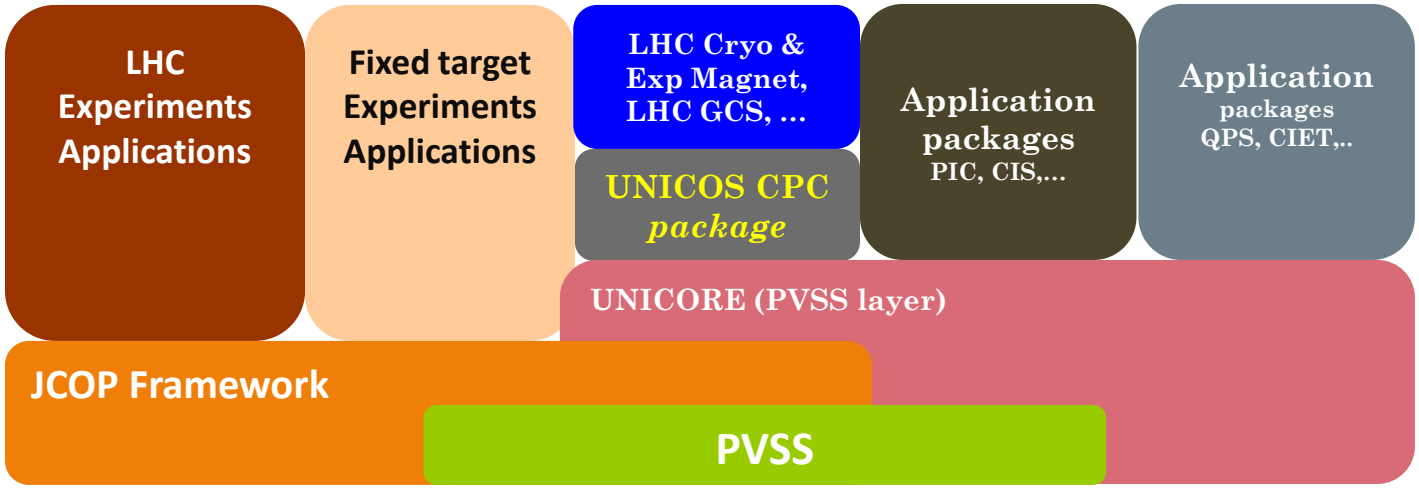
Magnet thermo-switches @ 60C





FRAMEWORKS INTEGRATION

- At the supervision level, UNICOS framework reuses some of the JCOP framework components



UCPC Process Control development

- **Standards** (or *best-practices*)

  ISA-88 / IEC-61512 / IEC-61499 : Batch control / Distributed systems
  - Open architecture for distributed control and automation, applying object oriented techniques to process control
  - Organize objects into a hierarchy (process/equipment hierarchy)
  - Well placed and separated code for the process control logic

- Process control systems design: **Object Orientation**
  - *Modularity*: optimize design, tests and maintainability
  - *Abstraction*: only interfaces seen, reduces complexity
  - *Generic*: Any module can be plug-in without major impact
  - *Re-use*: Minimize engineering effort

NOT ONLY A BUNCH OF DEVICES

- 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

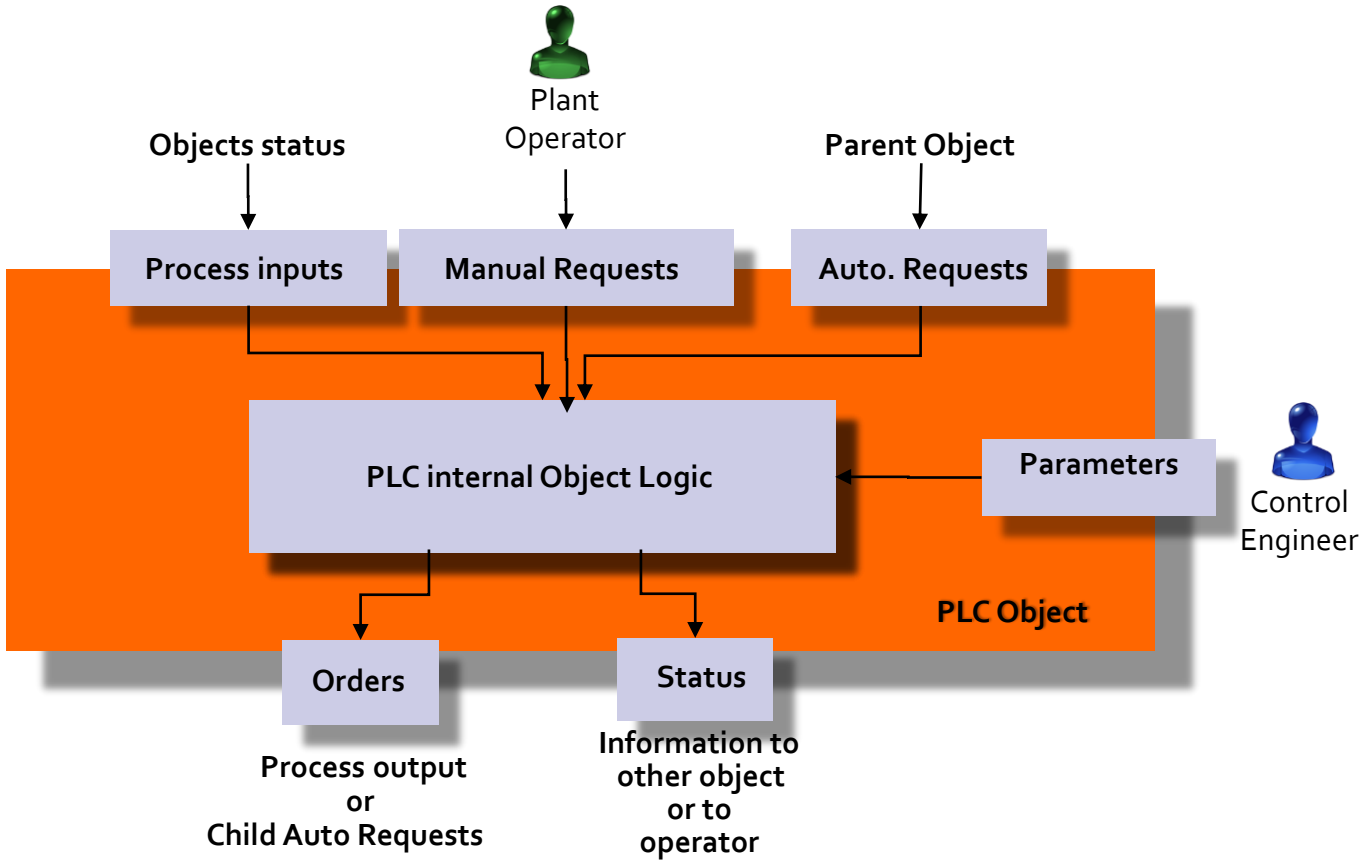
  - **Interface Objects**
    - ✓ Parameter (Digital, Word, Analog)
    - ✓ Status (Word, Analog)

- A **formalized** way of :
  - Define the **control units** of a process (ISA-88 standard: Batch processes)
  - **Programming** the **specific process logic** for those units

Outline

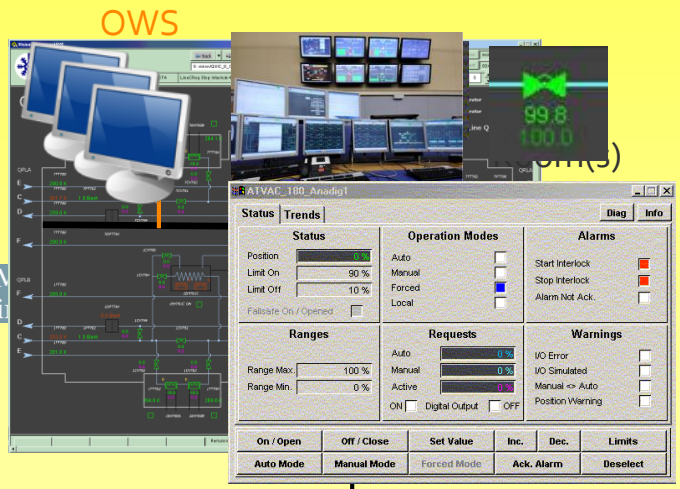
- UNICOS vs. UNICOS-CPC
- **Objects main functionality and connectivity**
  - **Objects**
  - **Relationships**
- From specs to implementation: Overview
  - Life cycle
  - Generation Tools

UNICOS CPC OBJECT MODEL

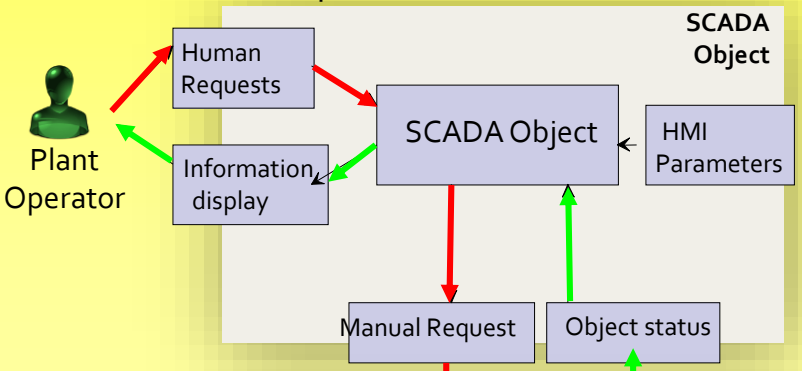


OBJECTS & LAYERS INTEGRATION

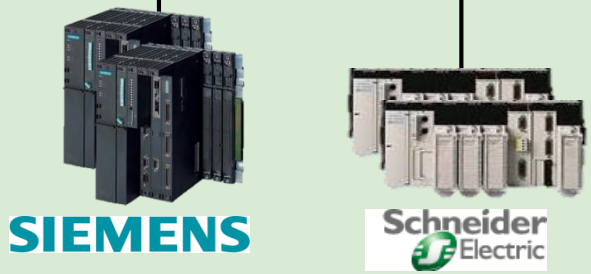
Supervision Layer

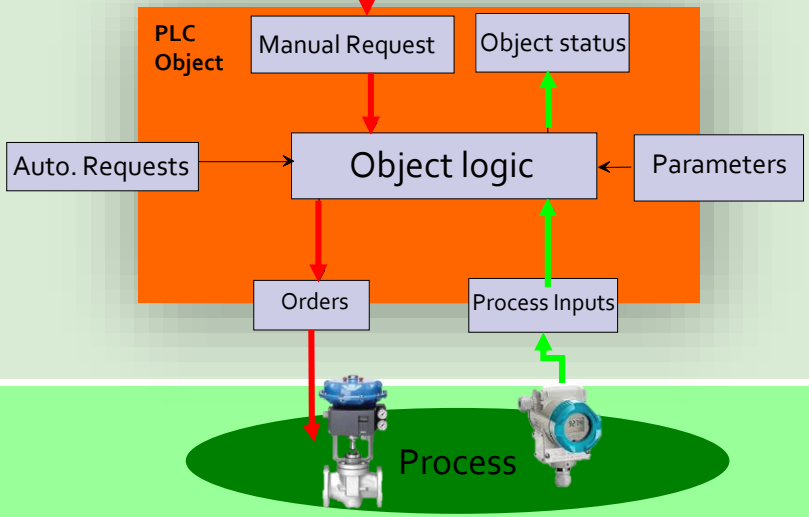


In the Supervision layer the object presents the relevant information to the operator and allow manual commands



Control Layer

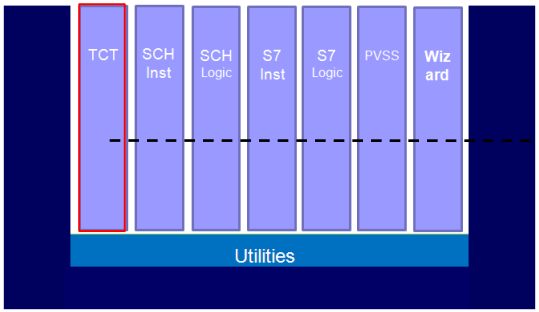


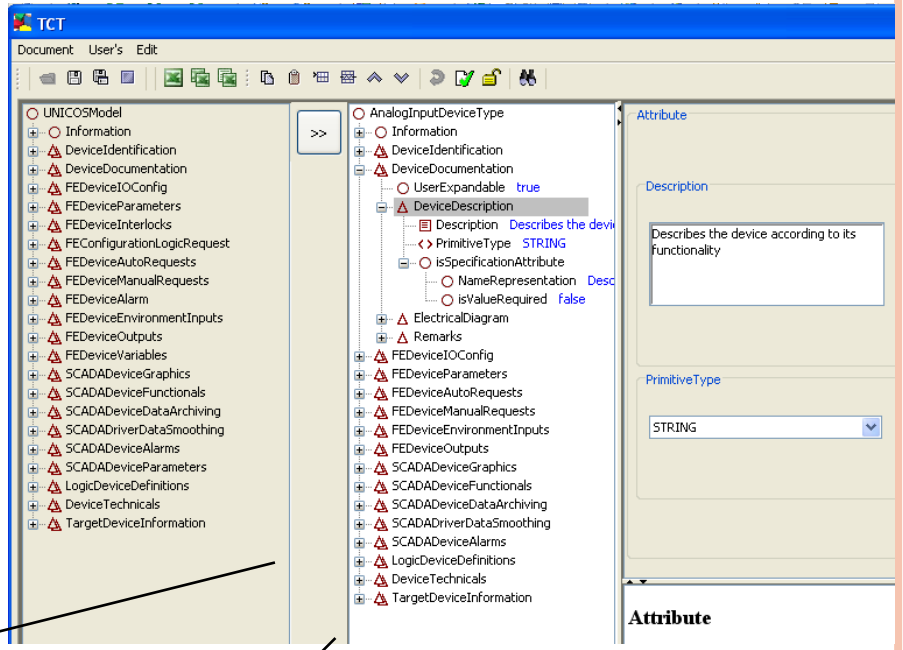


Field Layer

DEVICES CREATION: TCT

TCT: define the UNICOS Types





- The model is supported by a *meta-model* describing the properties of the model.

- TCT (Type Creation Tool) provides a *drag&drop*-based mechanism to build new device type definition described in XML

Specifications.xml

	A	B	C	D	E	F
1	Name	AnalogInput				
2	Object Type Family	IOObjectFamily				
3	Description	19/Jun/2010				
4	Version	1.6				
5	Version Comments	ted by the development team				
6	Status	UnderDevelopment				
7						
8	eviceIdentificatio	DeviceDocumentation				
9						
10	Name	Description	Electrical Diagram	Remarks	FE encoding type	InterfaceParam1
11						
12						
13						
14						
15						
16						
17						

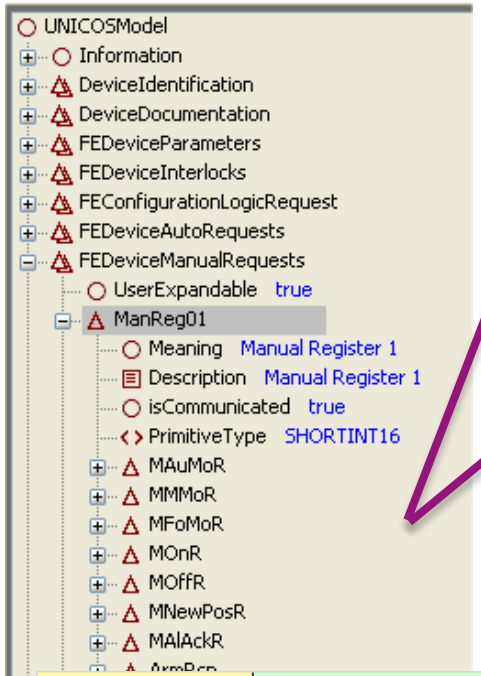
UNICOSTypeDefinition.xml

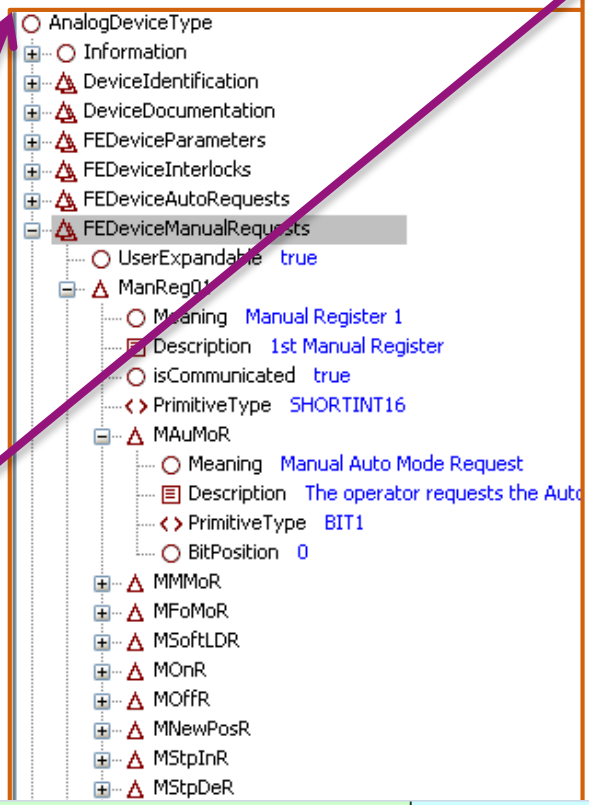
Type documentation

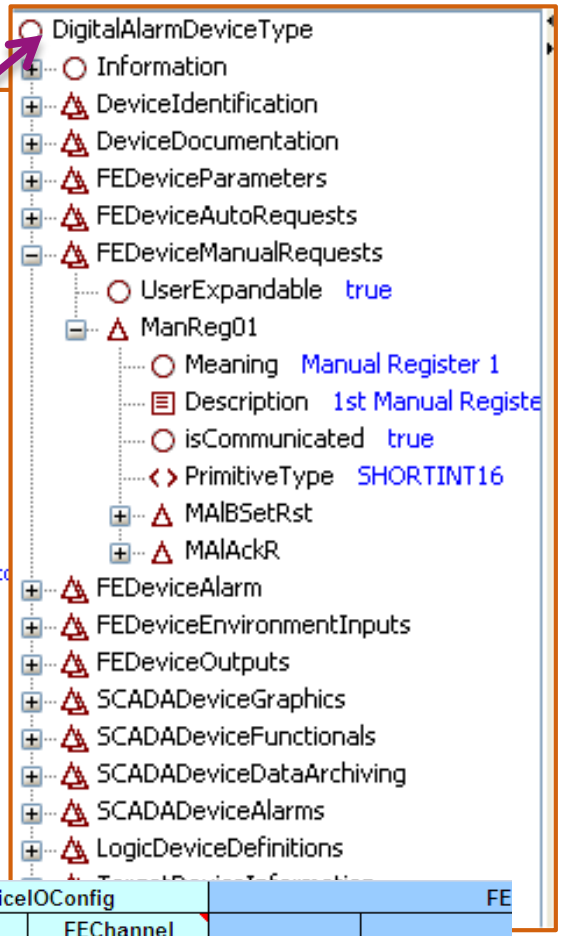
EN/ICE Web page

TCT (Type Creation Tool)

UNICOS model







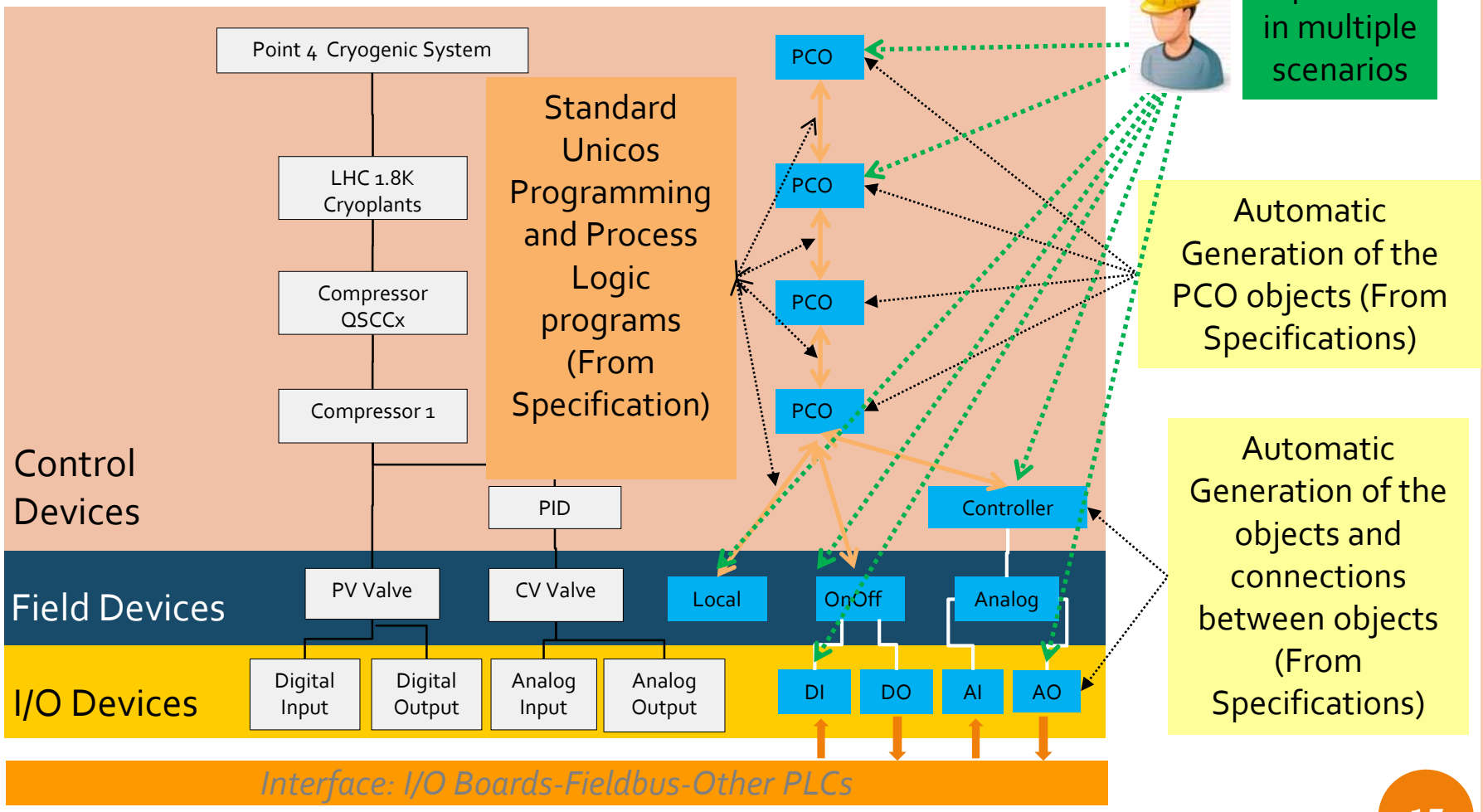
DeviceIdentification	DeviceDocumentation			FEDeviceIOConfig		FE	
					FEChannel		
Name	Description	Electrical Diagram	Remarks	FE encoding type	InterfaceParam1	Range Min	Range Max
QSDN_4_1TT4001	Vessel 1- Heater section1-Temp. control	AI1.0			%IW1.1.0	80	350
QSDN_4_AI1	SPARE	AI1.1			%IW1.1.1	0	100
QSDN_4_1TT4002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

UNICOS-CPC OBJECTS

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

PROCESS VS. CONTROL

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

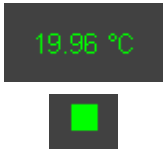


I/O OBJECTS

- **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



Field Objects

- **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

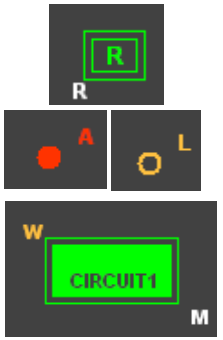
CONTROL OBJECTS

- **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:
    - Explicitthreshold: 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



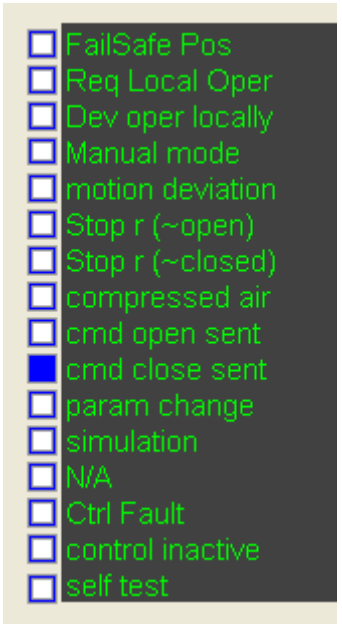
Interface Objects

- **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



UNICOS-CPC Fundamentals

- **Operation Modes**
- **Alarms & Interlocks**
- **Controllers**
- **Recipes**

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.

Alarms & Interlocks (1)

- **Interlock**: Asynchronous condition carrying an actuator or a unit to its safety position (or preventing from starting). An interlock must not be used for normal operation but for abnormal behaviour. UCPC (soft) interlocks are not guarantying functional safety !

- The possible interlocks for a complete unit or for an actuator are:
  - **Full Stop Interlock (FS)**: Stop the unit/actuator (all dependent units/actuators are set to their fail-safe position) and wait manual acknowledgement before restarting.
  - **Temporary Stop Interlock (TS)**: Stop the unit/actuator (all dependent units/actuators are set to their fail-safe position) and restart automatically when the interlock disappears.
  - **Start Interlock (SI)**: Prevent the unit from starting (all dependent units/actuators stay in their fail-safe position).

- **Alarm (AL)**: It is an indication of a potential problem to aware operator in SCADA.
- *Alarm and Interlock are only one object with different functionality*

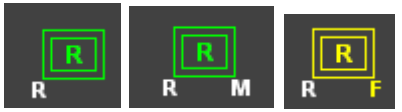
Alarms & Interlocks (2)

- **Acknowledgement**
  - Can be done in advance while the alarm is still active
  - Can be directly performed to a unit/actuator, then, all dependent alarms of this unit/actuator are automatically acknowledged by propagation.
- **Mask/Block**
  - Operators can mask alarms in the UI. The alarm is still active in the PLC !
  - Operators can block alarms in the UI. The alarm is then blocked in the PLC, then no subsequent actions due to that alarm (if any) !!!!
- **Resume Full stop on units/actuators.**

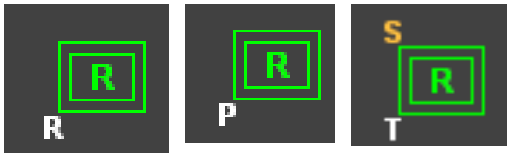
  The operator have two options to resume the unit (the unit will resume only once the interlock is gone):
  - By ACK the interlock
  - By ACK and "allow restart" action explicitly. This can be done even while the interlock is active (if configured in the specifications)

CONTROLLERS

- UNICOS Mode management as other field objects
  - **A**uto/**M**anual/**F**orced with same meanings



- Working States
  - **R**egulation
  - **P**ositioning
  - **T**racking



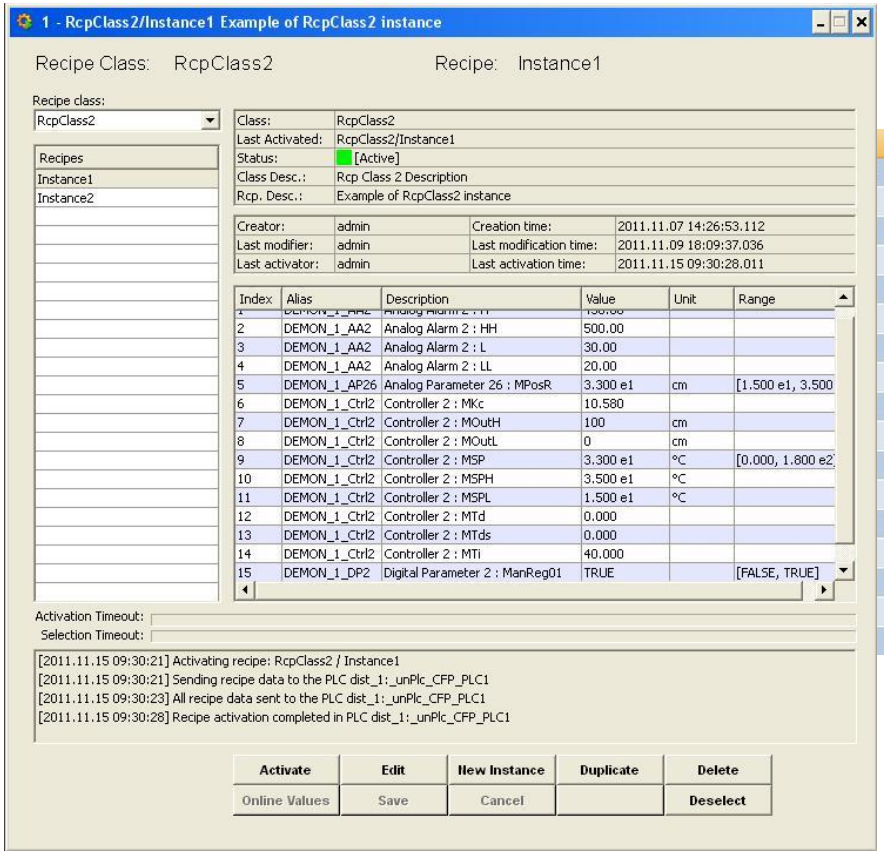
- PID features:
  - 1st order filter on Measured Value
  - PID Cycle time
  - Scaling mode (No Scaling / Input Scaling / IO Scaling)

- PID cascade automatically generated

DeviceIde	FEDeviceParameters		
	Controller Parameters		
Name	MV Filter Time (s)	PID Cycle (s)	Scaling Method
HRM_876_TC0001	0.1	0.1	No Scaling
HRM_876_PID1	0.02	0.05	Input Scaling
HRM_876_PID2	0.1	0.1	Input/Output Scaling

PLANT OPERATION: RECIPES

- Allows bulk parameterization and/or multiple setting in one shot
- Established list of recipe-able objects (e.g. PID parameters, Xpar for Thresholds,…)



Example of a HMI recipe tool (creation, activation, edit,...)

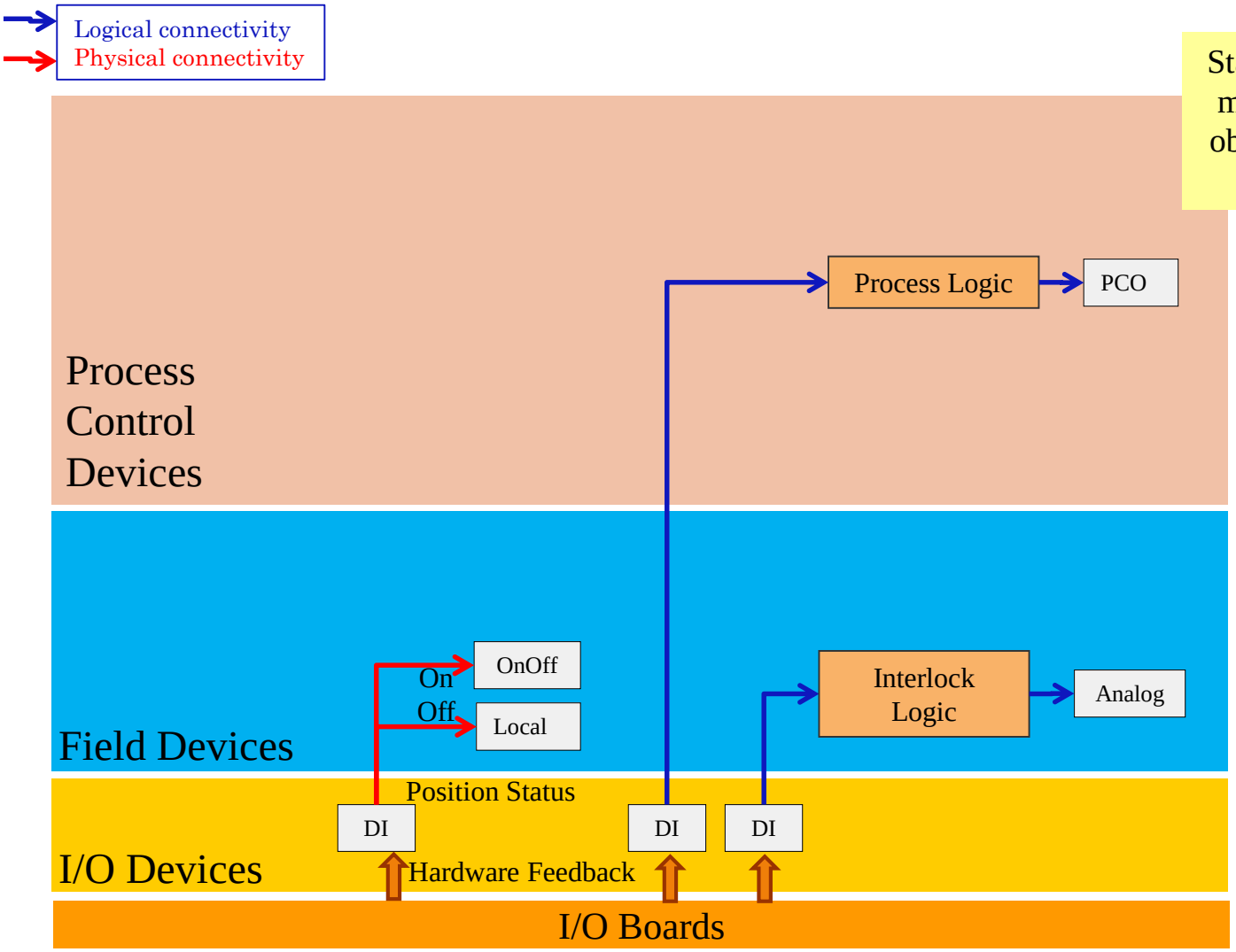
initialRecipe	deviceType	deviceAlias	dpe	value	unit	range
TRUE	-					
	DigitalParameter	DEMON_1_DP2	ManReg01	TRUE		
	WordParameter	DEMON_1_WP2	MPosR	25	bar	[0, 100]
	AnalogParameter	DEMON_1_AP26	MPosR	33	cm	[15, 35]
	AnalogAlarm	DEMON_1_AA2	HH	500	°C	[0, 220]
			H	450	°C	[0, 220]
			L	30	°C	[0, 220]
			LL	20	°C	[0, 220]
	Controller	DEMON_1_Ctrl2	MSP	33		
			MSPH	35		
			MSPL	15		
			MOutH	100		
			MOutL	0		
			MKc	10.58		
			MTi	40		
			MTd	0		
			MTds	0		

Example of a recipe instance

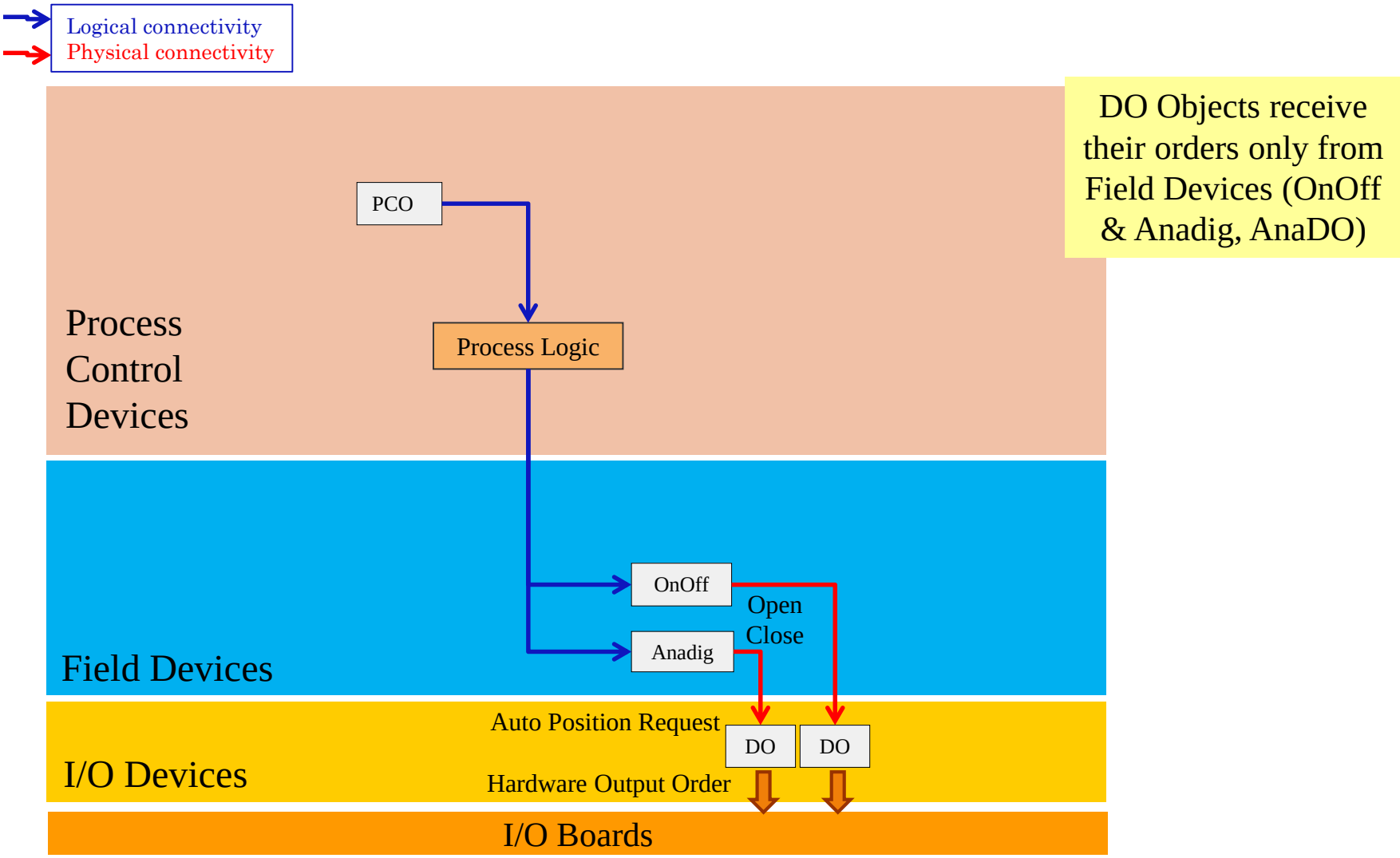
UNICOS-CPC Connectivity

- **Connectivity between objects**
- **Relationships when programming**

DI Object Connectivity

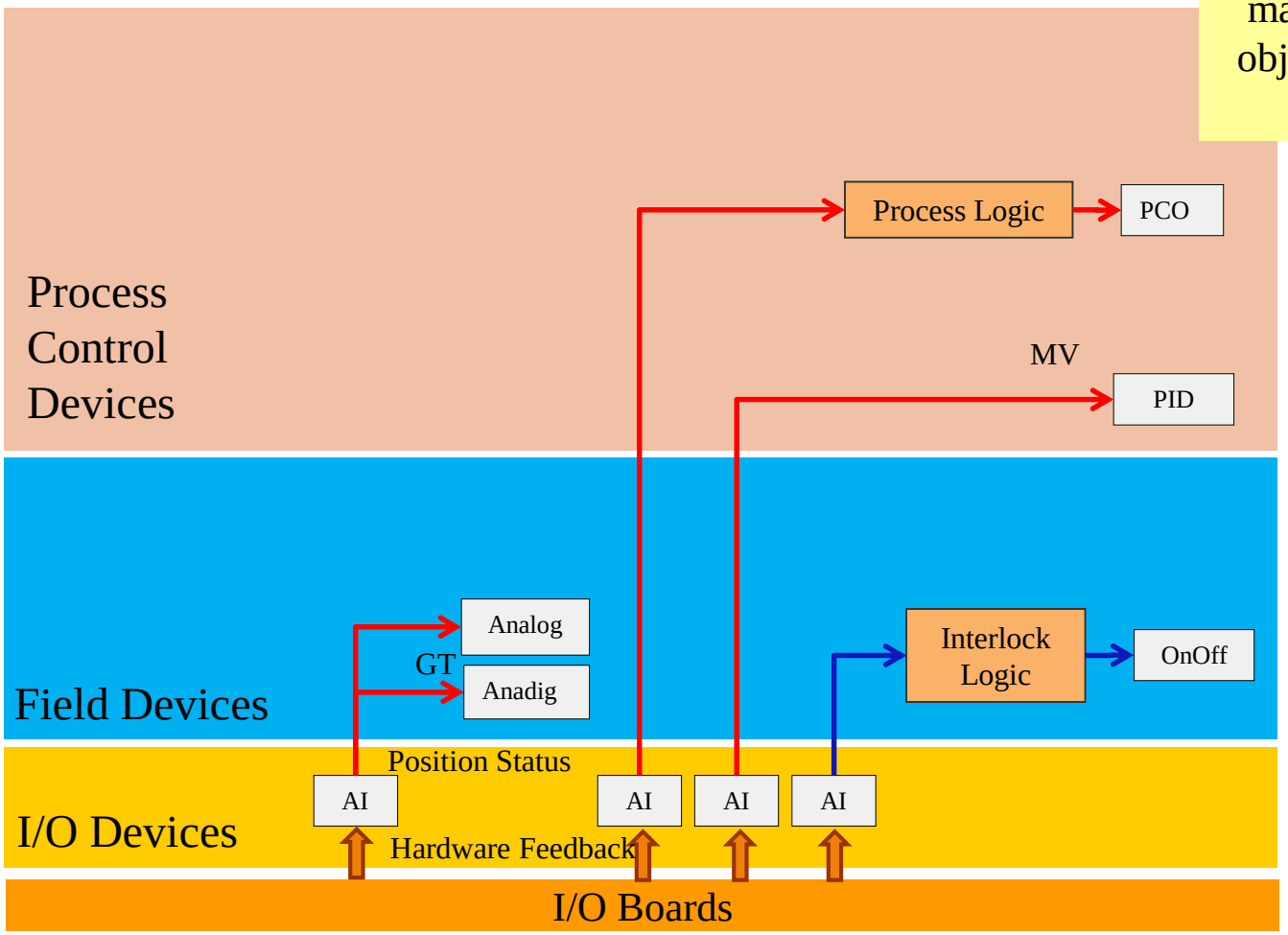


DO Object Connectivity

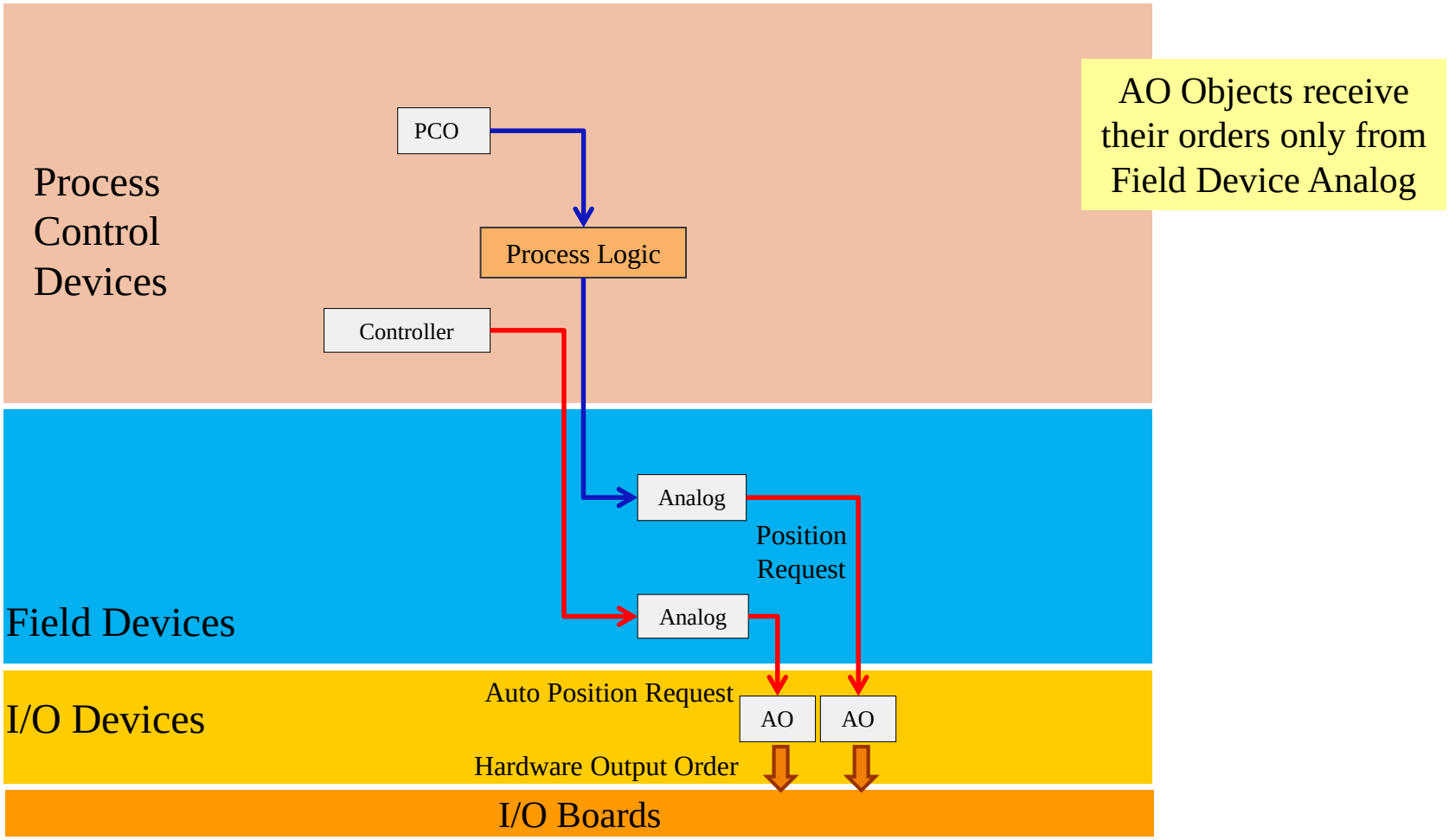


AI Object Connectivity

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

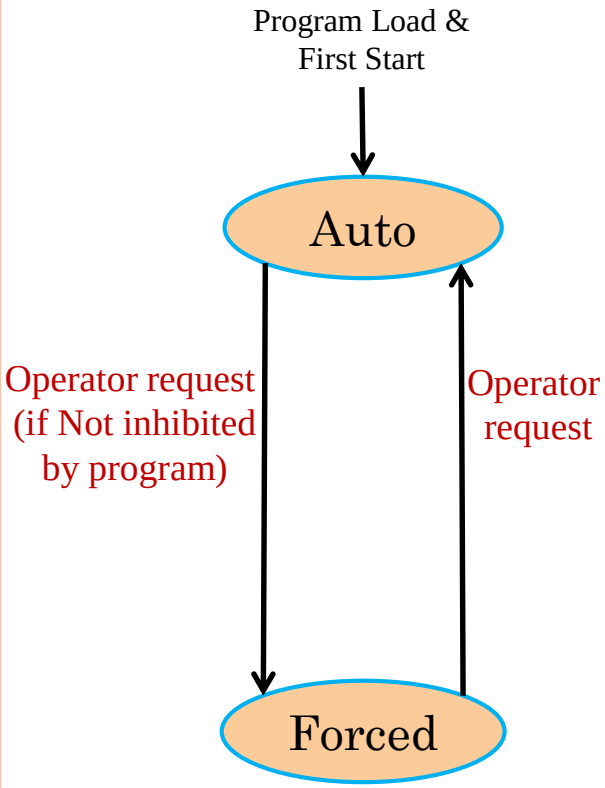


AO Object Connectivity



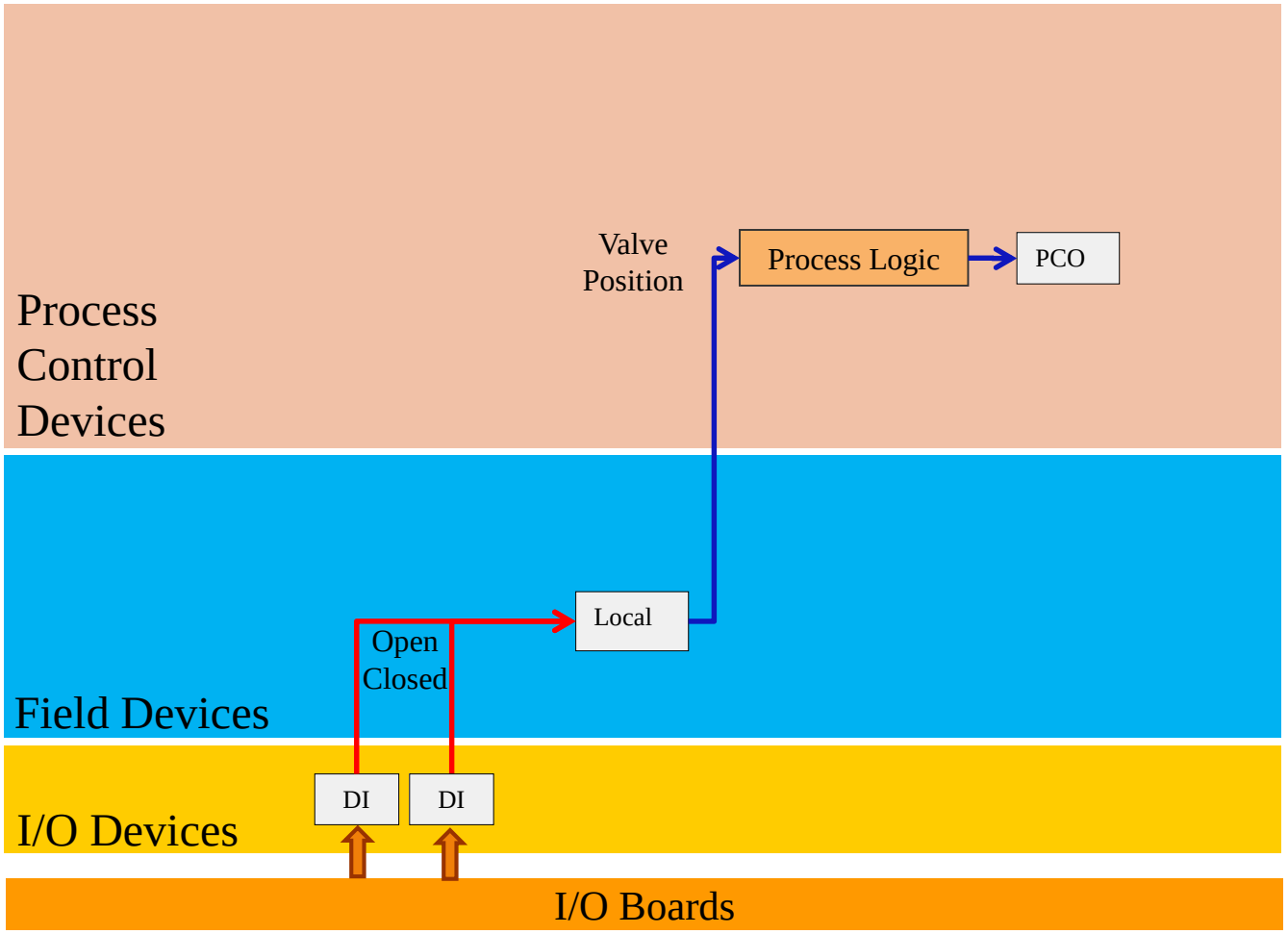
AO Objects receive their orders only from Field Device Analog

Forced Mode and Position Status of I/O Objects



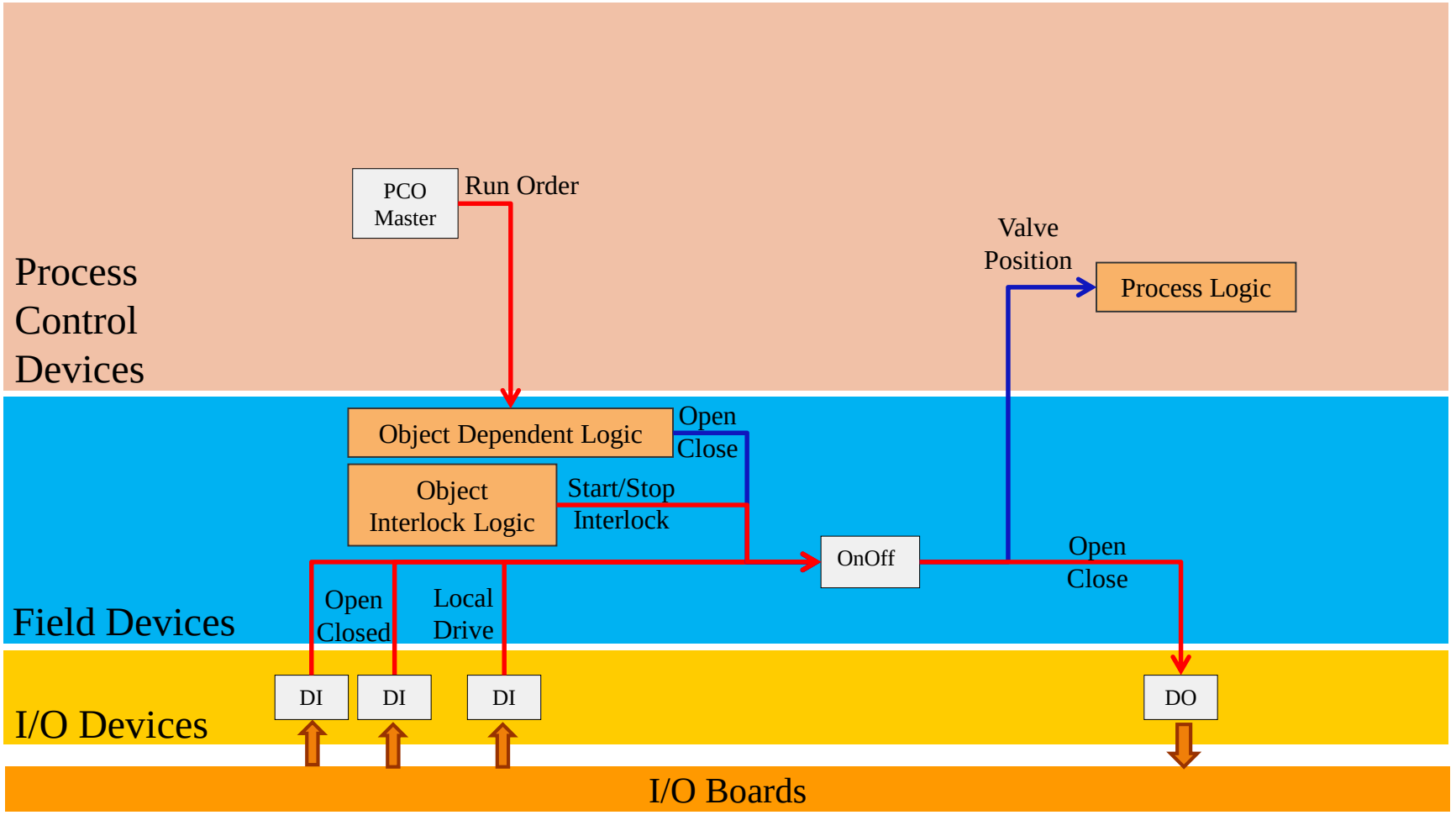
- Forced Mode for operators ("**expert**" privilege):
  - For AI & DI Objects, when the operator forces the value, the Position Status may become different from the Hardware Feedback.
  - For DO & AO Objects, when the operator forces the object, the Output Order sent to the process may be different from the Auto Position Request of the process.
- Forced Mode may be useful for:
  - Electrical tests
  - Program simulation
  - Degraded operation!

Local Object Connectivity

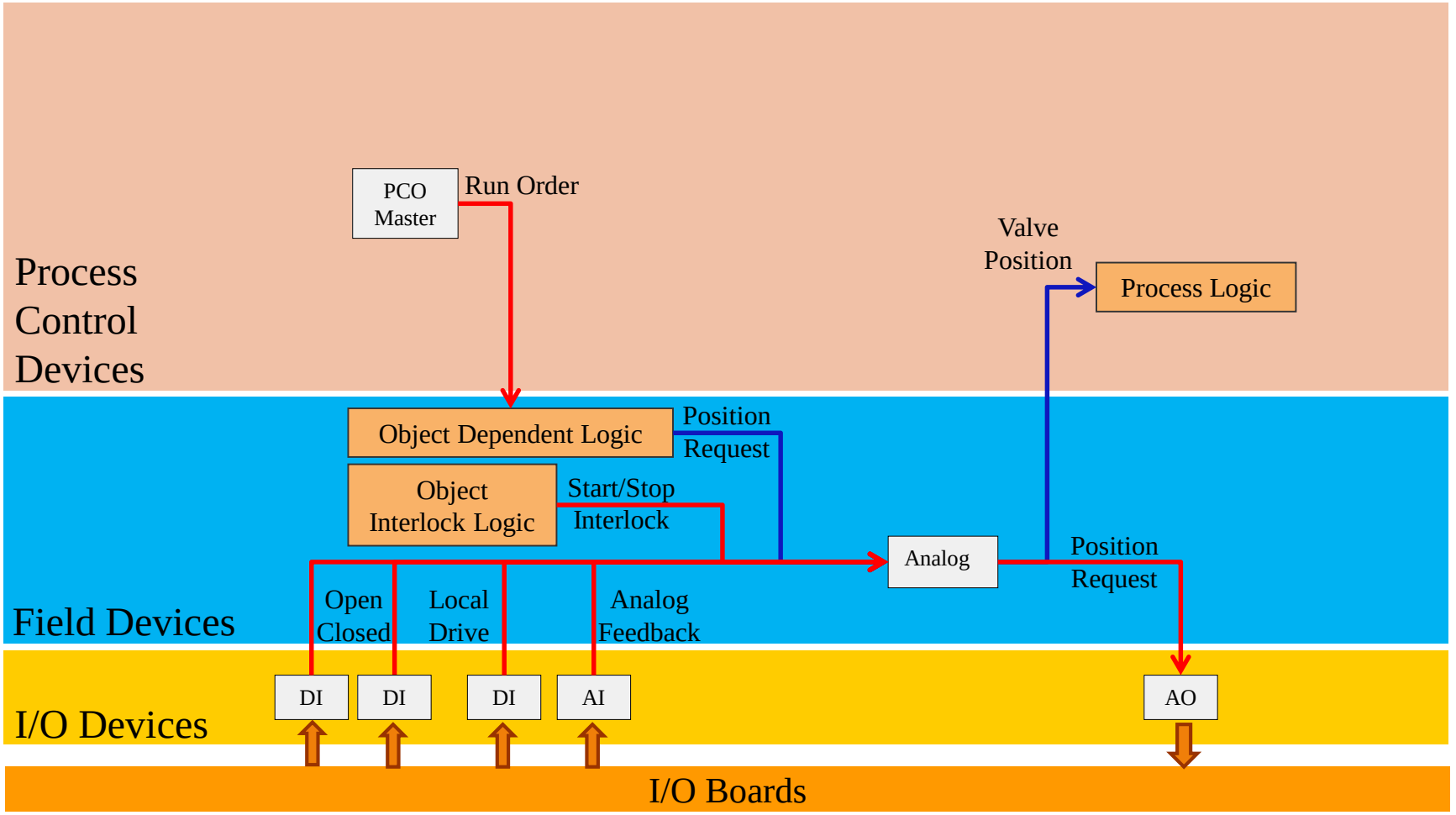


Status of Local Objects may be used by all objects anywhere in the program.

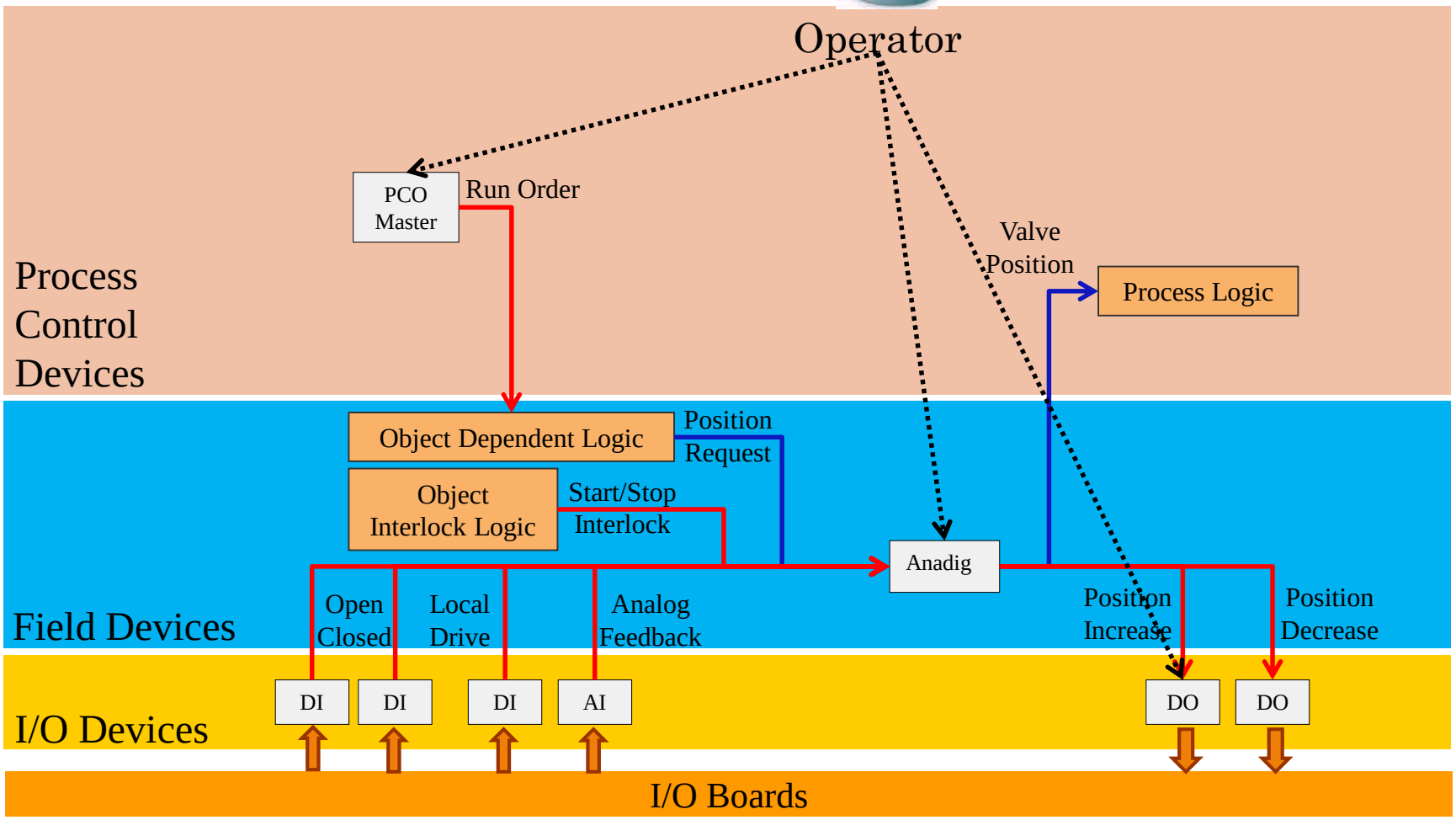
OnOff Object Connectivity



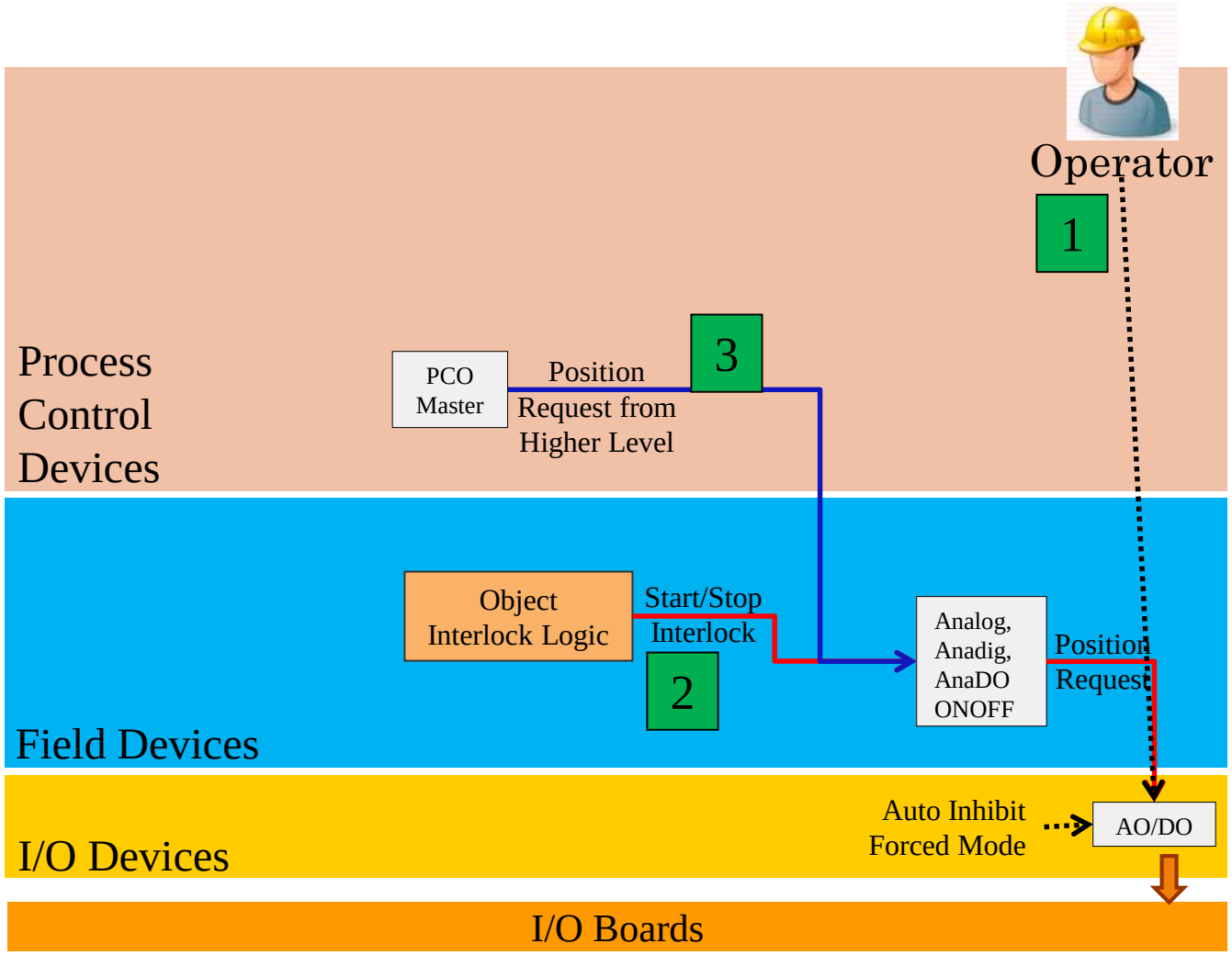
Analog Object Connectivity (no PID)



ANADIG OBJECT CONNECTIVITY (NO PID)



What are the priorities?
Example with AO/DO

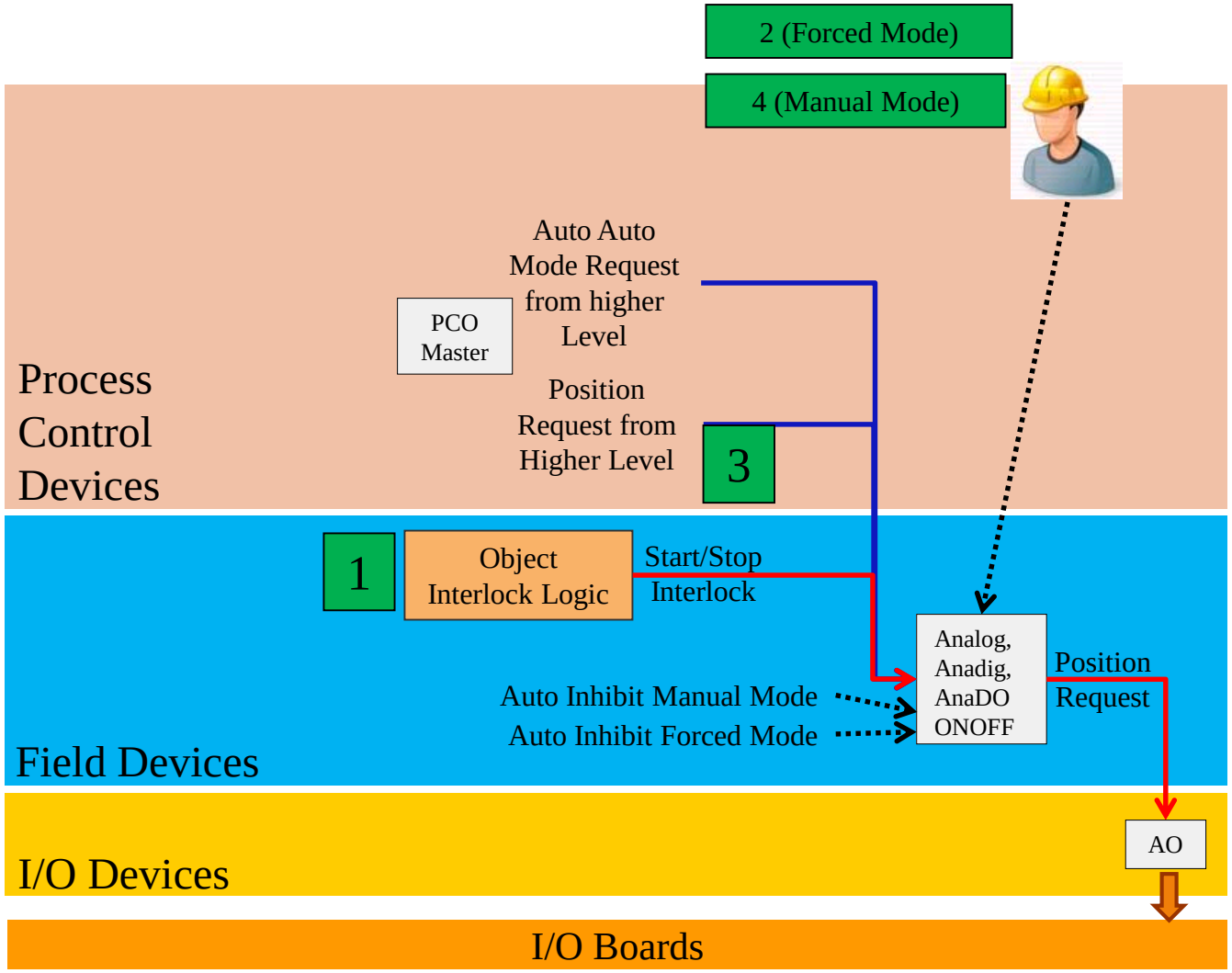


X → priority order

Operator takes Forced Mode and forces the AO/DO Objects to a position.

- This action overrides all the logic (safety) !!
- Possibility to inhibit the forced mode by program parameter

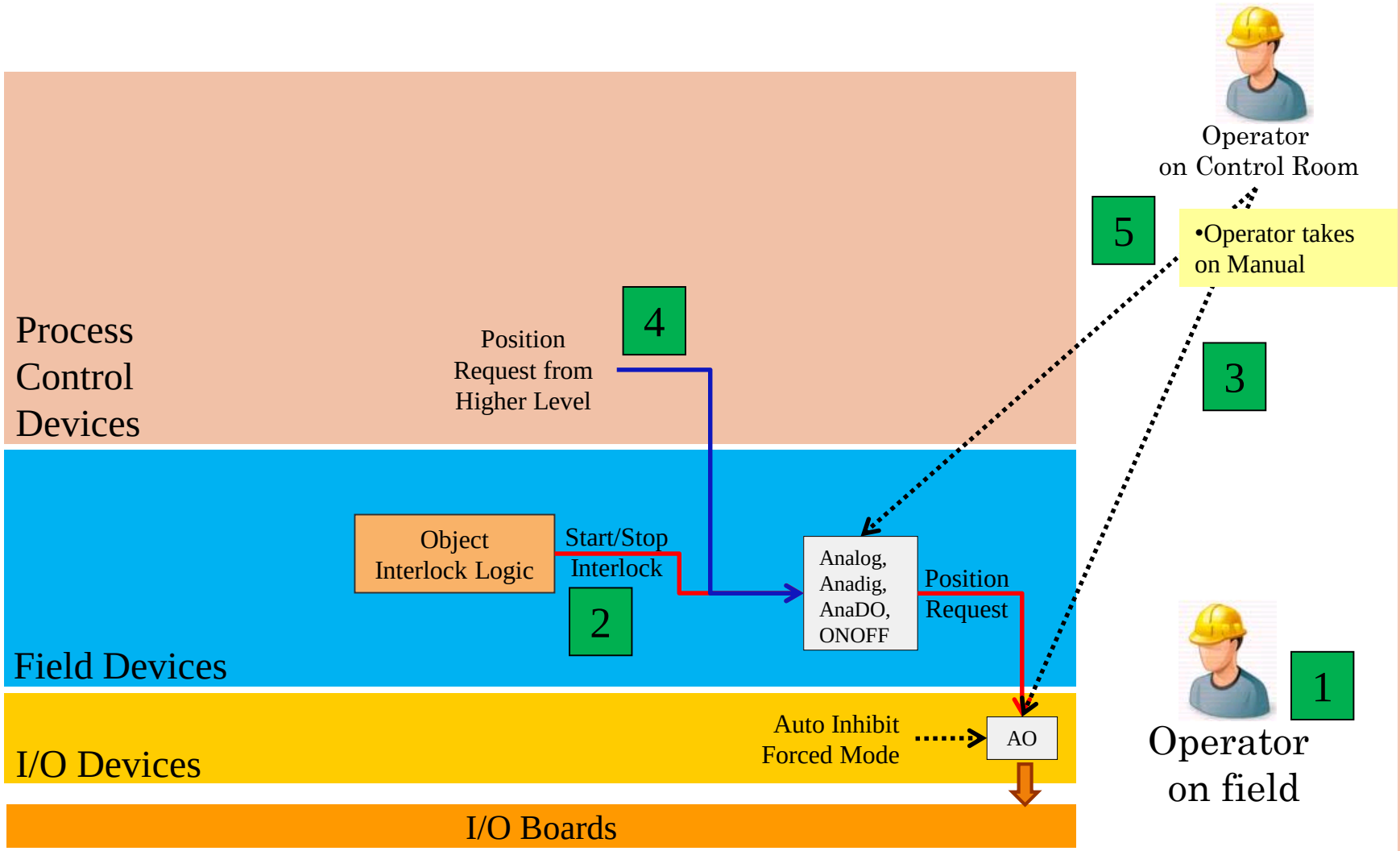
WHAT ARE THE PRIORITIES?



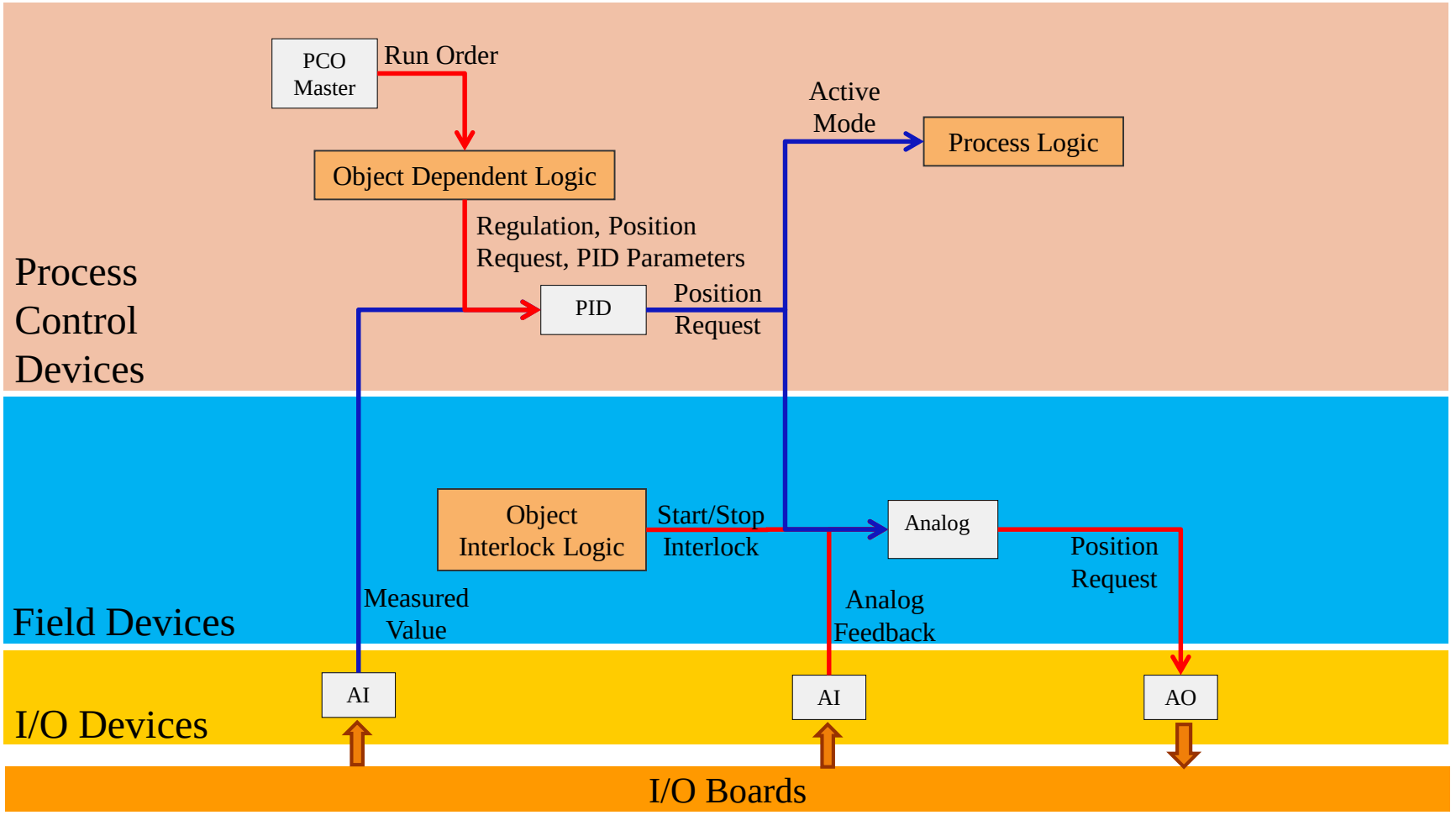
Operator takes Manual or Forced Mode and sets the Field Objects to a position.

- This action overrides the position request coming from higher level, but the interlock logic has higher **priority**.
- Auto Mode may be requested by Higher level if Manual Mode
- Possibility to inhibit the forced/manual mode by program parameter

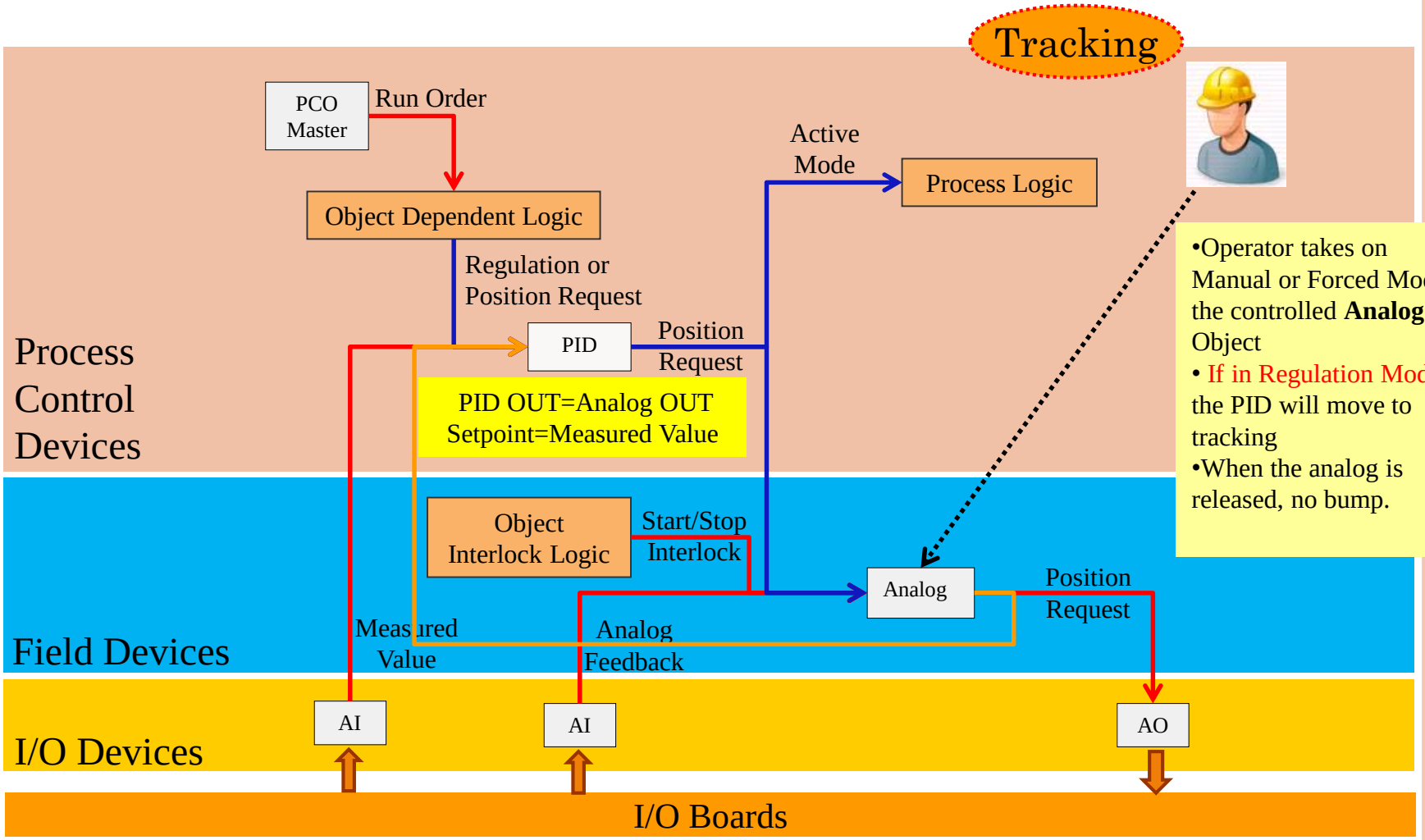
OPERATOR ON FIELD IS STRONGER



Controller Object Connectivity



CONTROLLER OBJECT TRACKING



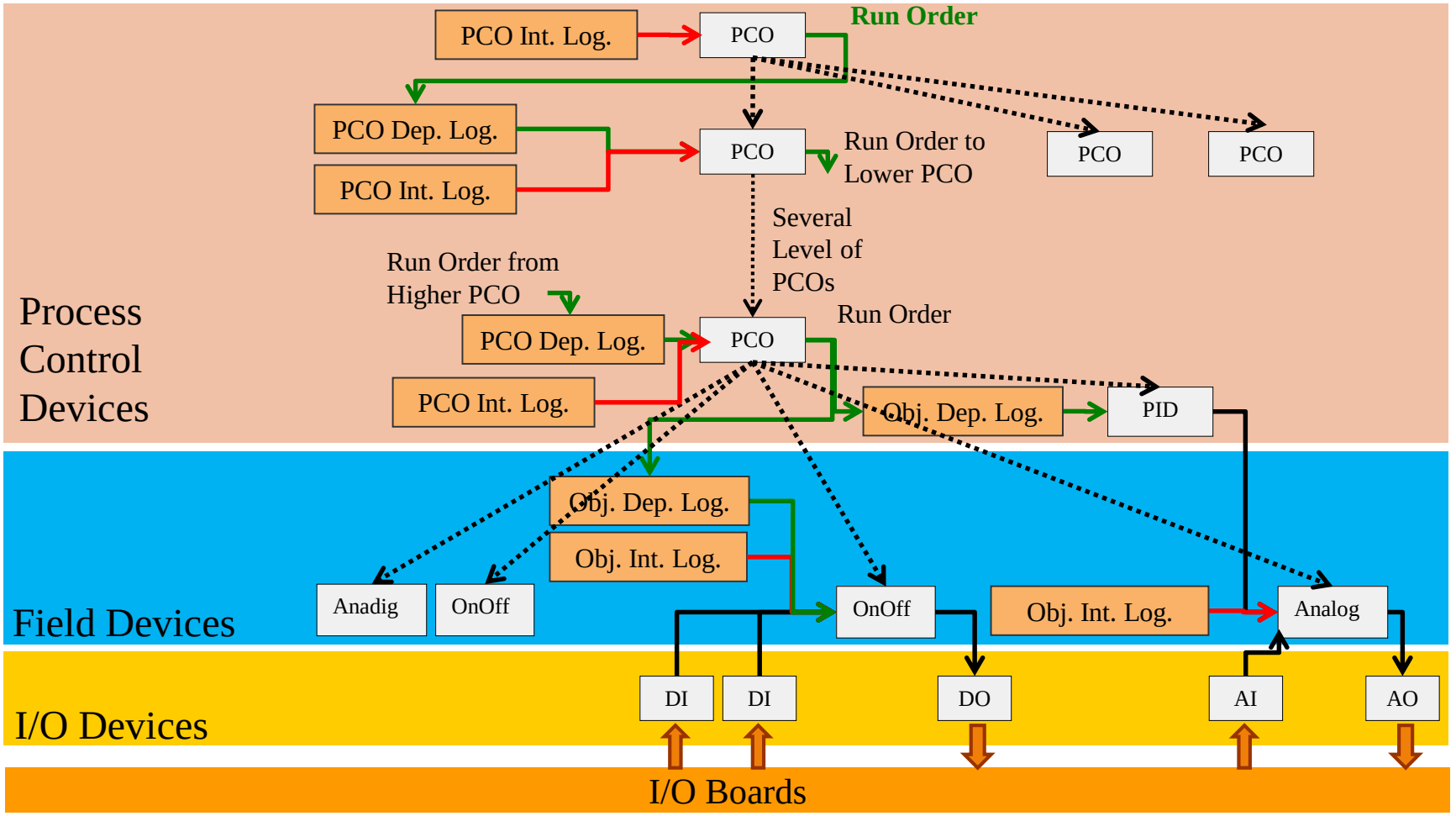
What's a PCO?

- PCO may be considered as a large OnOff object with some additional functionalities :
  - Full Stop Interlock (as field objects)
  - Block Alarm (Deactivate Interlocks Logics!)
  - Control Stop
  - Options (operation mode)

- PCO is used to represent sets of fields objects that performs a common function:
  - Compressor, Turbines
  - Vacuum System

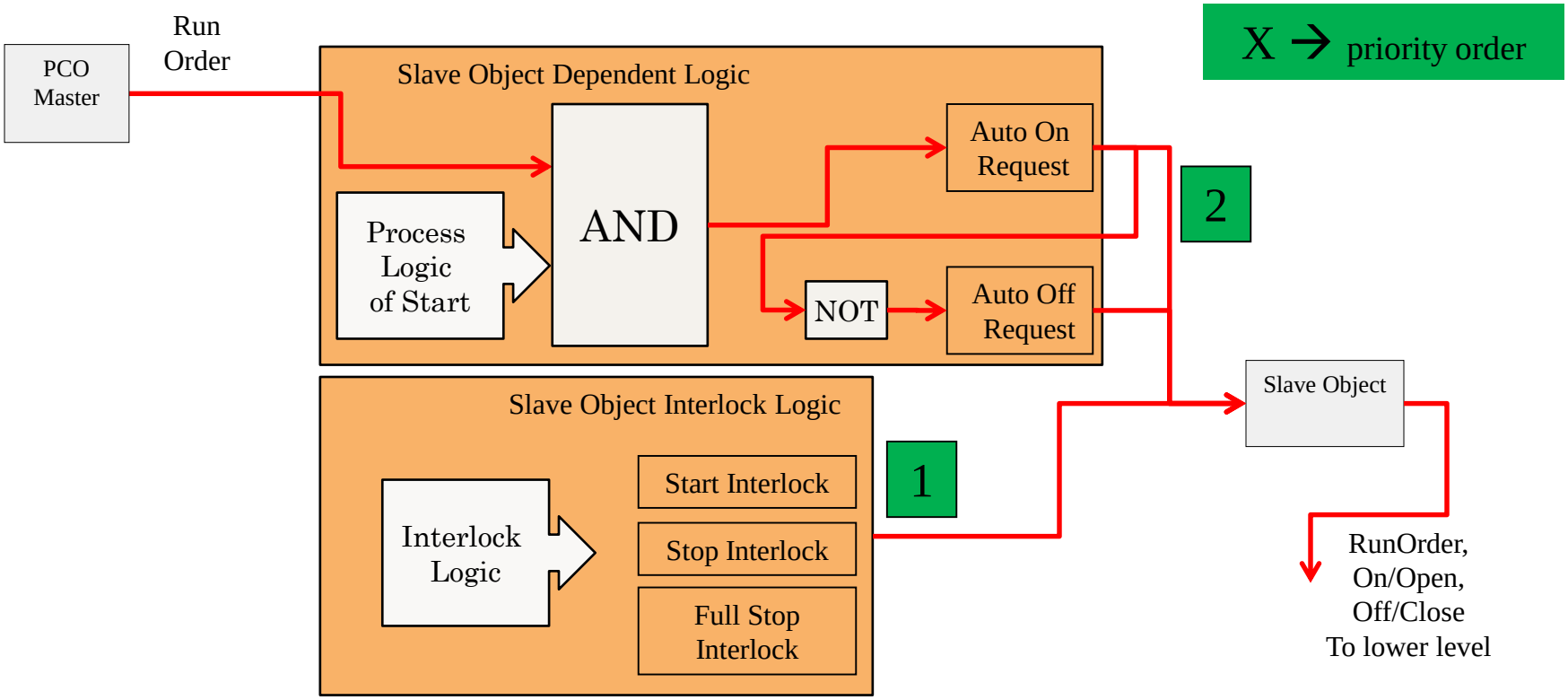
- PCO is also used to represent sets of other PCOs, this allows to manipulate large process parts:
  - Cold Box
  - Set of compressors

PCO Object Connectivity

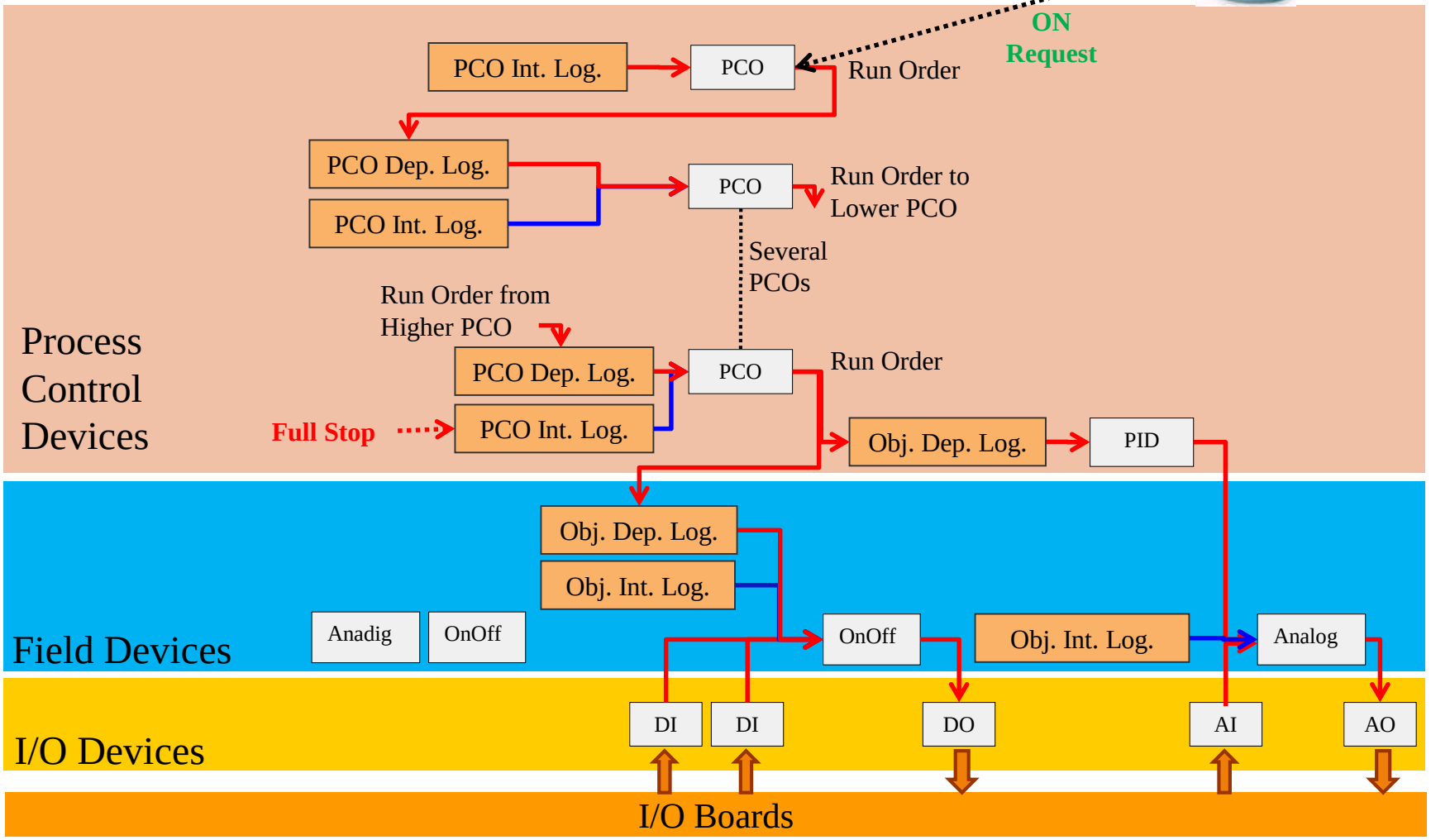


Run Order, Dependent Logic & Interlock Logic

- Run Order is the Start Request of the PCO to all the dependent objects

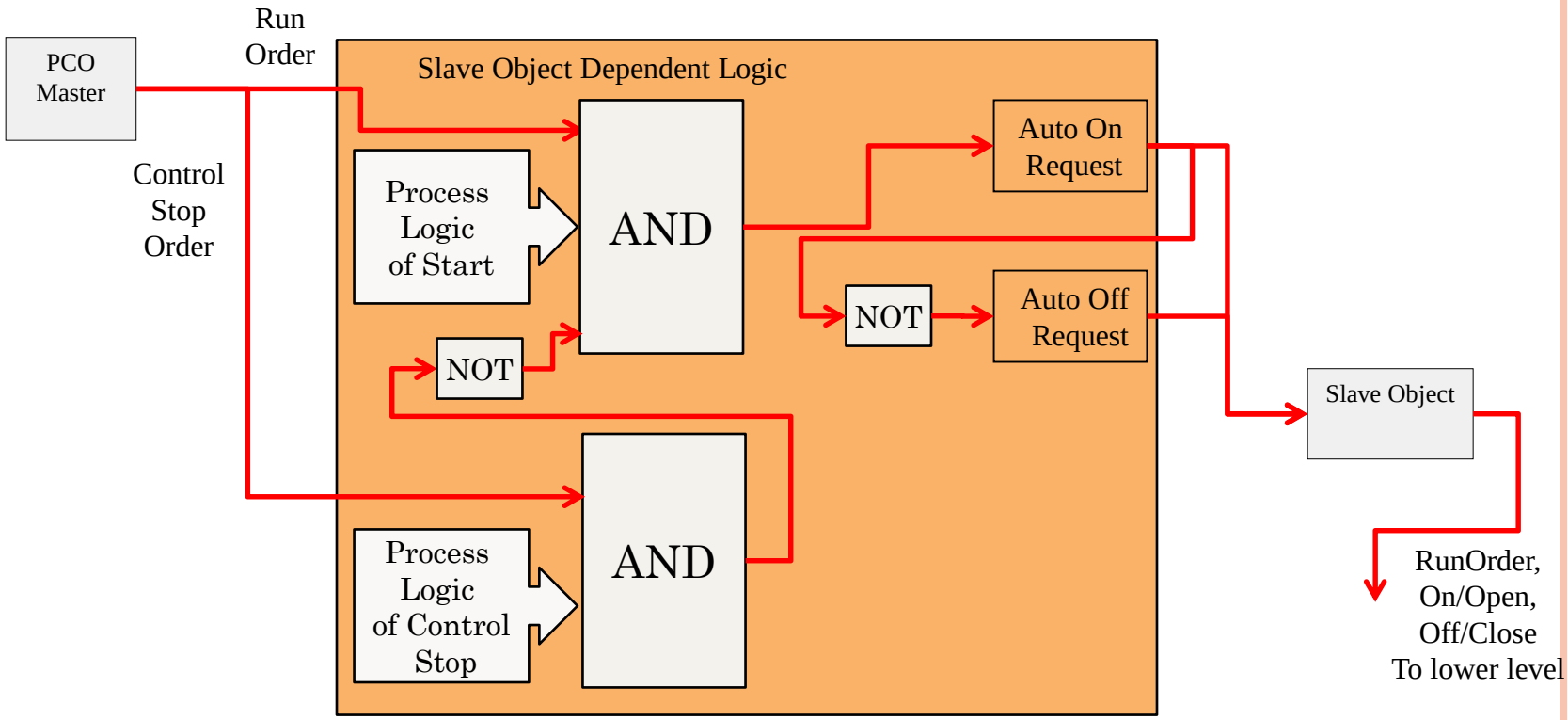


PCO Run Order propagation

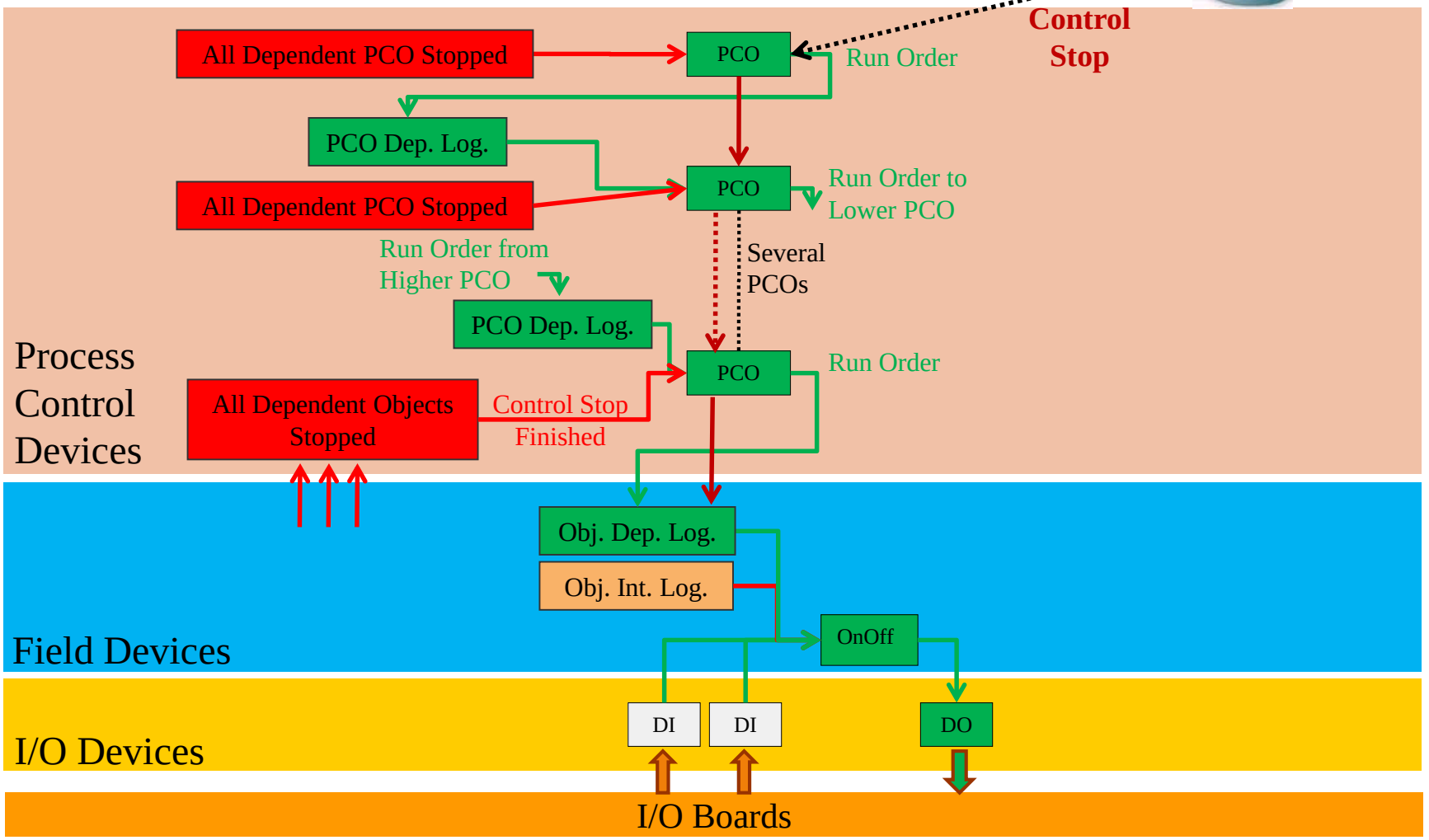


PCO Control Stop Order

- The control stop order is used to initiate a sequenced stop
- When a PCO receives a *control stop*, the RunOrder remains **ON** until the Control Stop ends

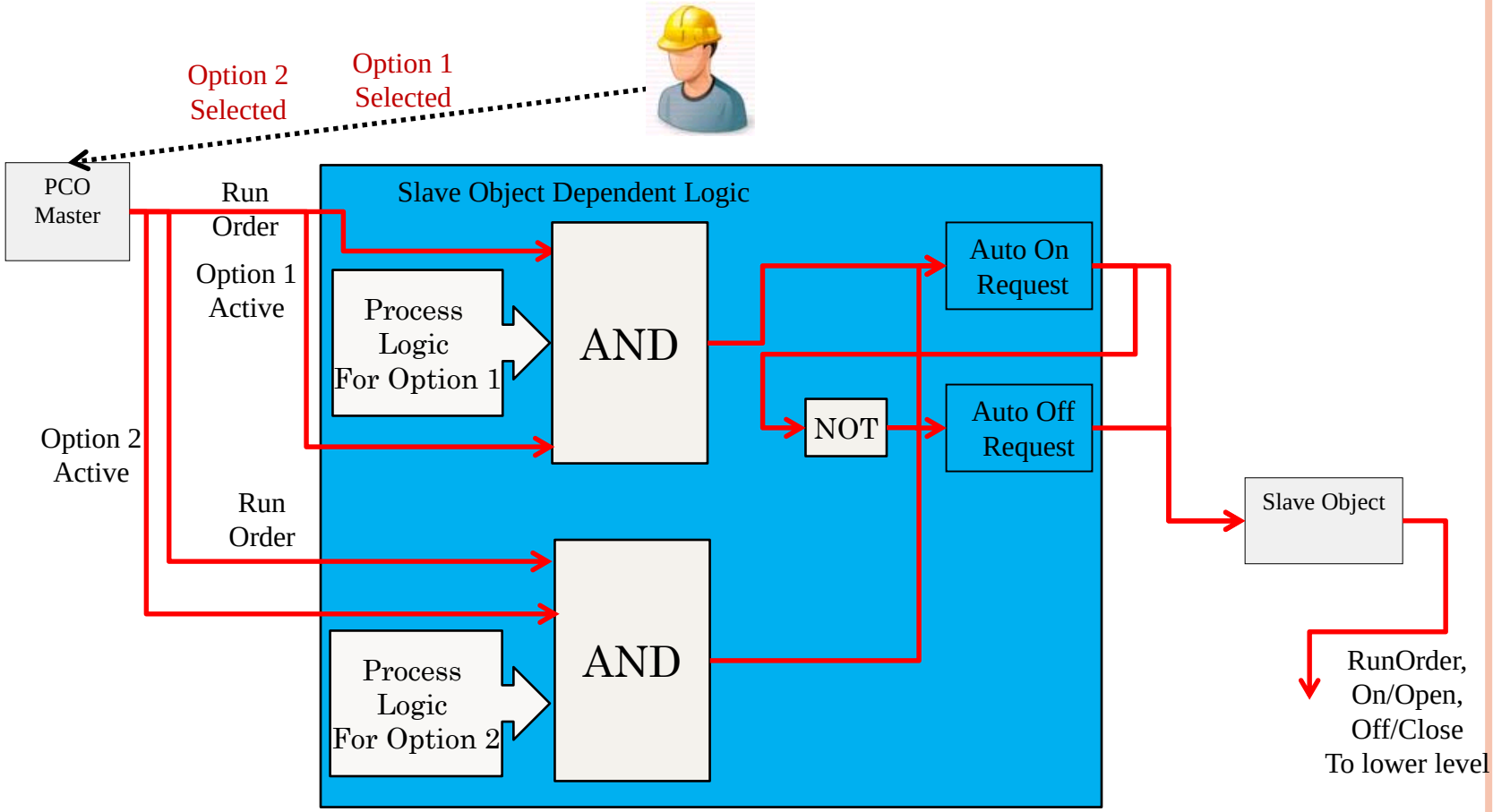


CONTROL STOP ORDER PROPAGATION



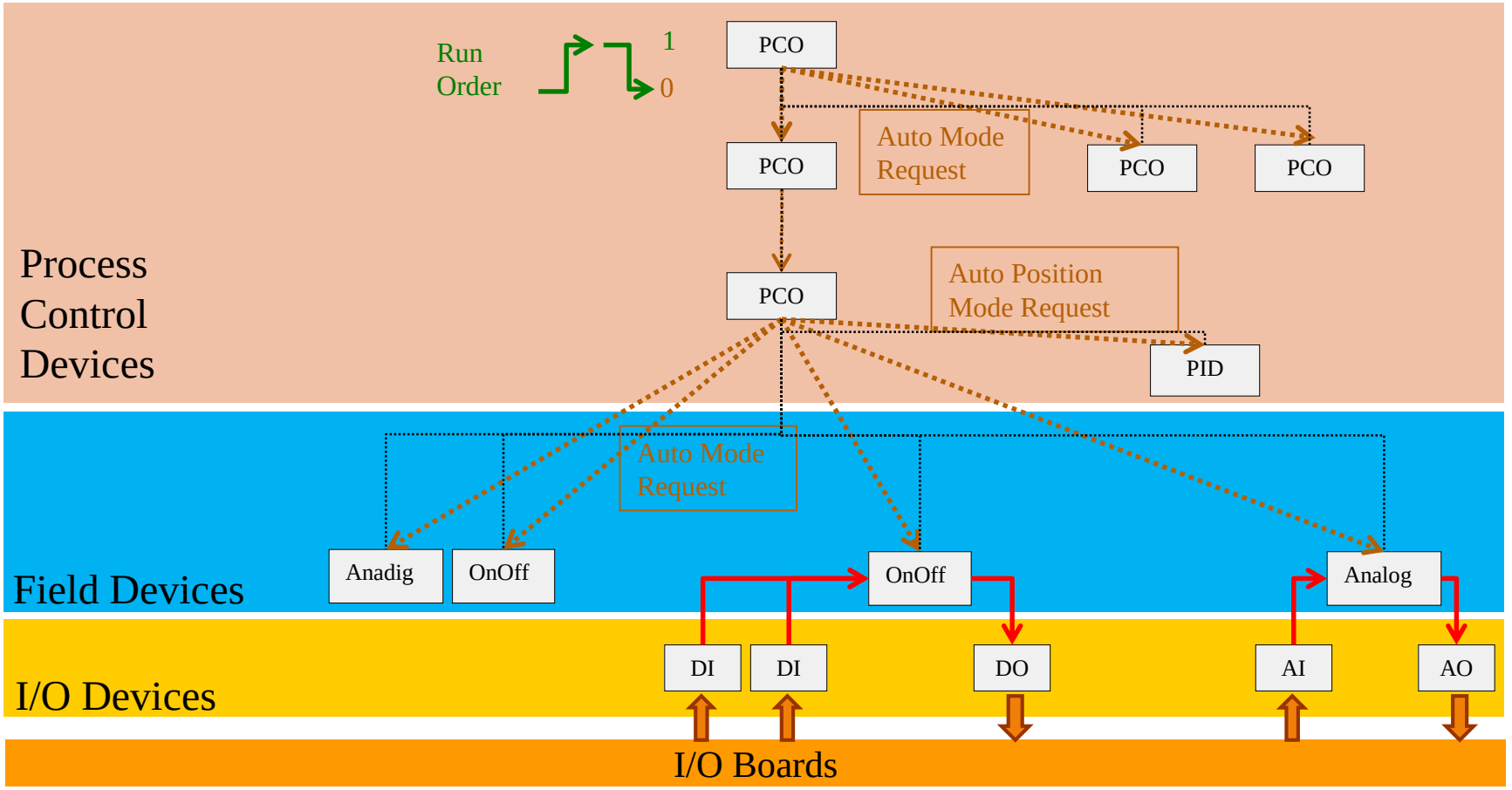
PCO Option

- The PCO Option allows the implementation of separate sets of logic. The option is selected by the operator.



Auto Mode Propagation

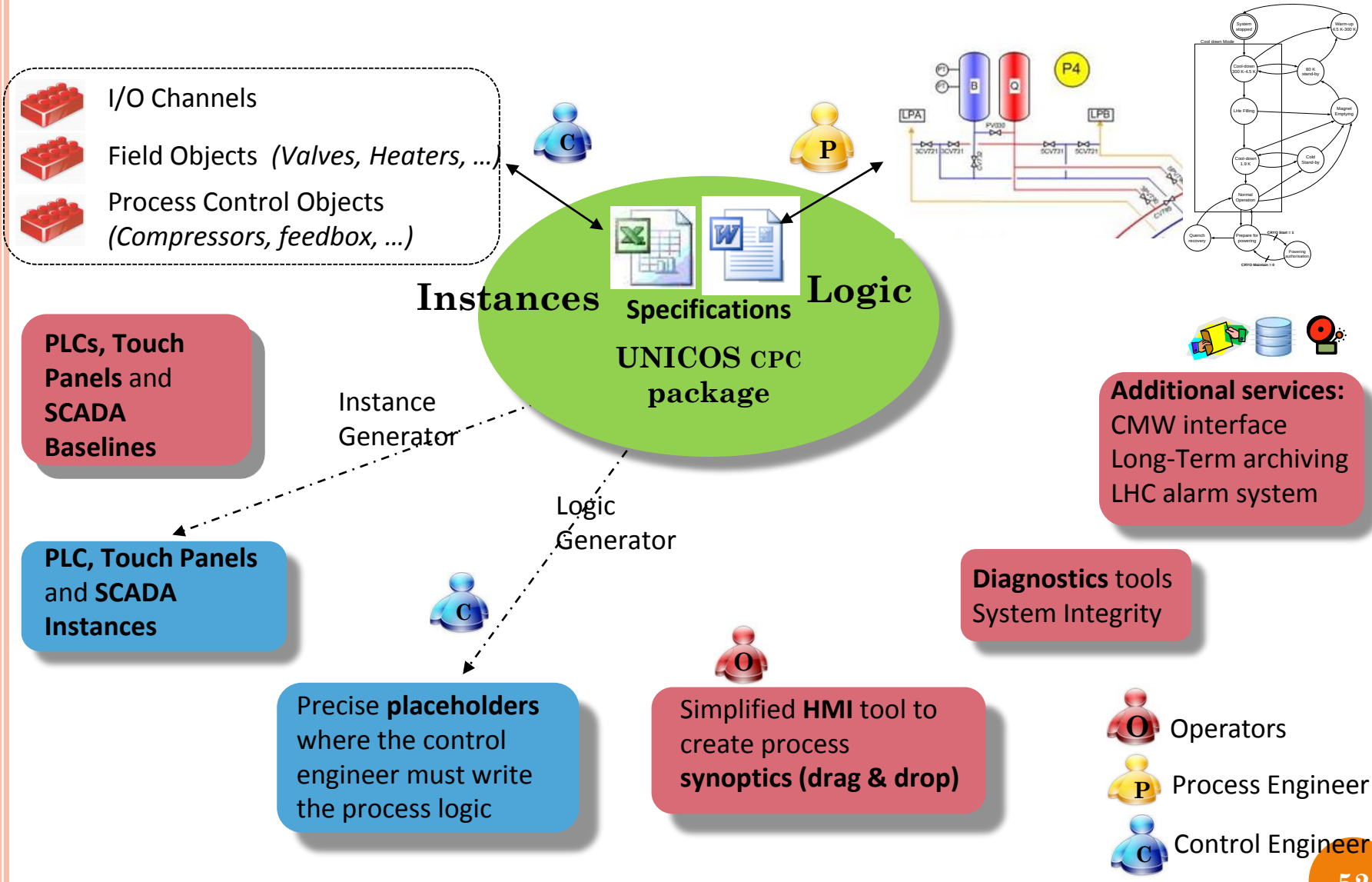
- When a PCO Object is started or stopped, all dependent objects are requested to Auto Mode



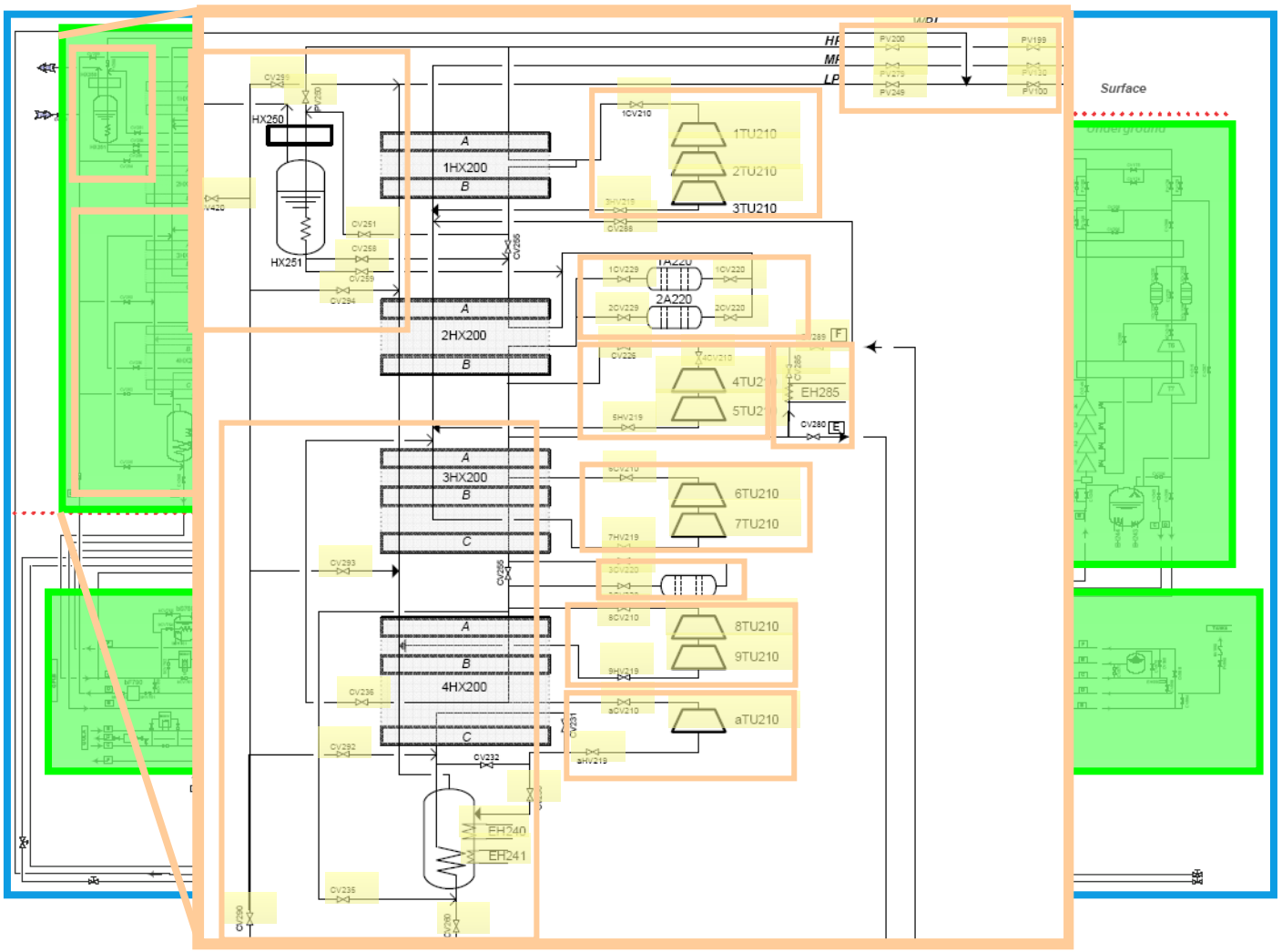
Outline

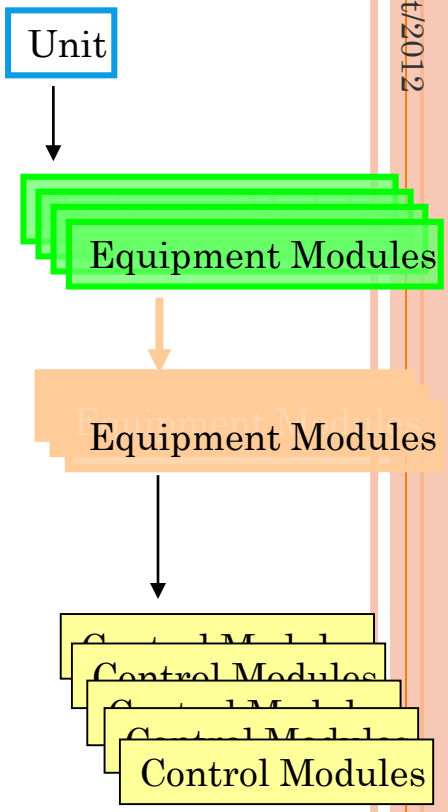
- UNICOS vs. UNICOS-CPC
- Objects main functionality and connectivity
- **From specs to implementation: Overview**
  - Life cycle
  - Generation Tools

UNICOS CPC OVERVIEW



DECOMPOSITION METHOD





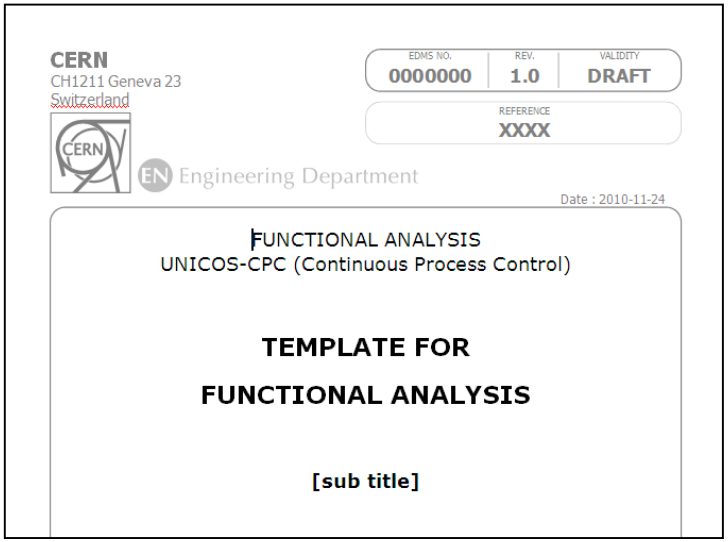
IEC 61512-1
Physical model

SPECIFICATIONS

DeviceIdentification	DeviceDocumentation			FEDeviceIOConfig		FEDeviceParameters				
					FEChannel					
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

EN Engineering Department

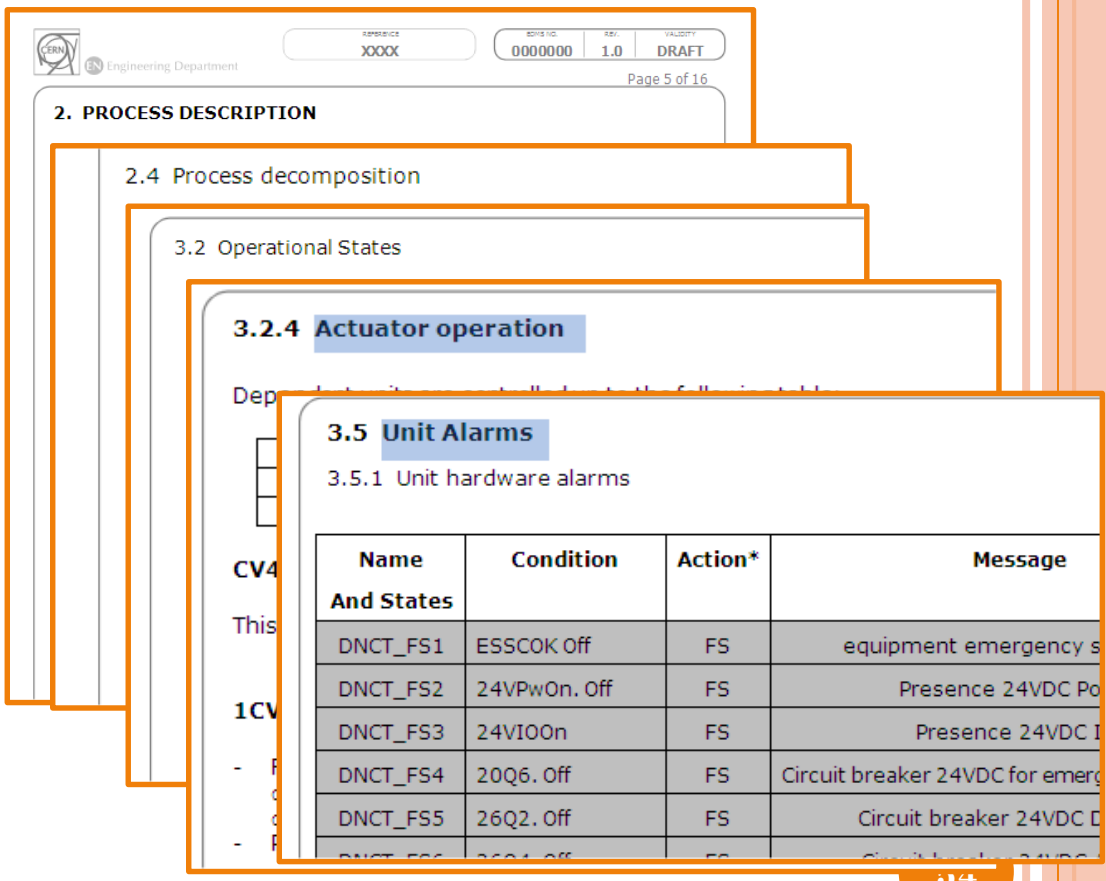
Date : 2010-11-24

FUNCTIONAL ANALYSIS
UNICOS-CPC (Continuous Process Control)

TEMPLATE FOR

FUNCTIONAL ANALYSIS

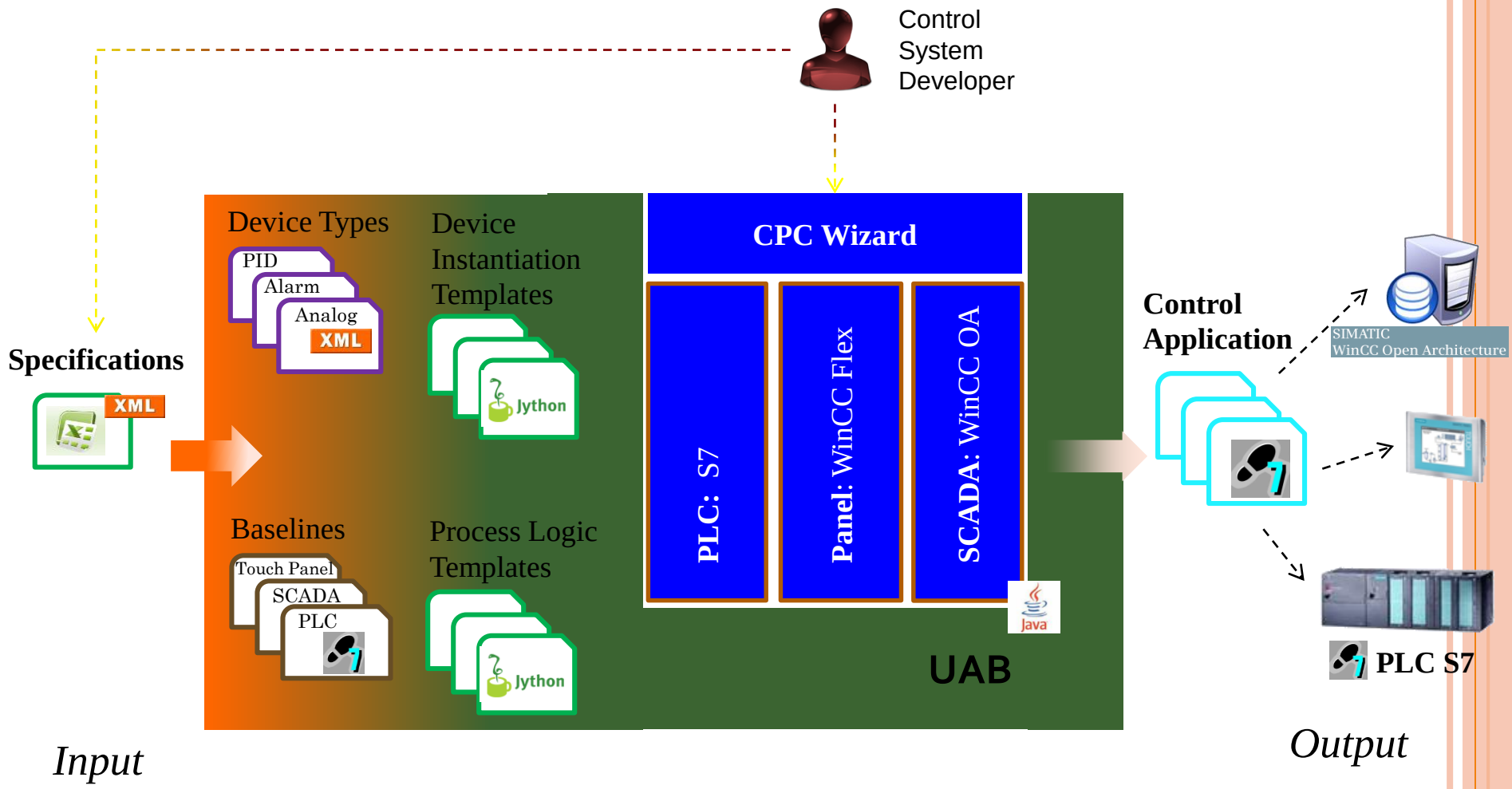
[sub title]



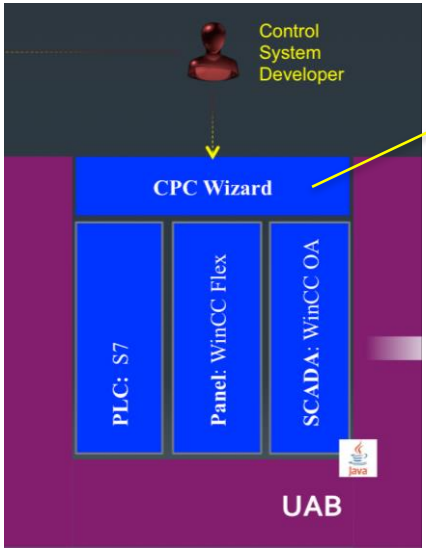
Code Generation Tools: Requirements

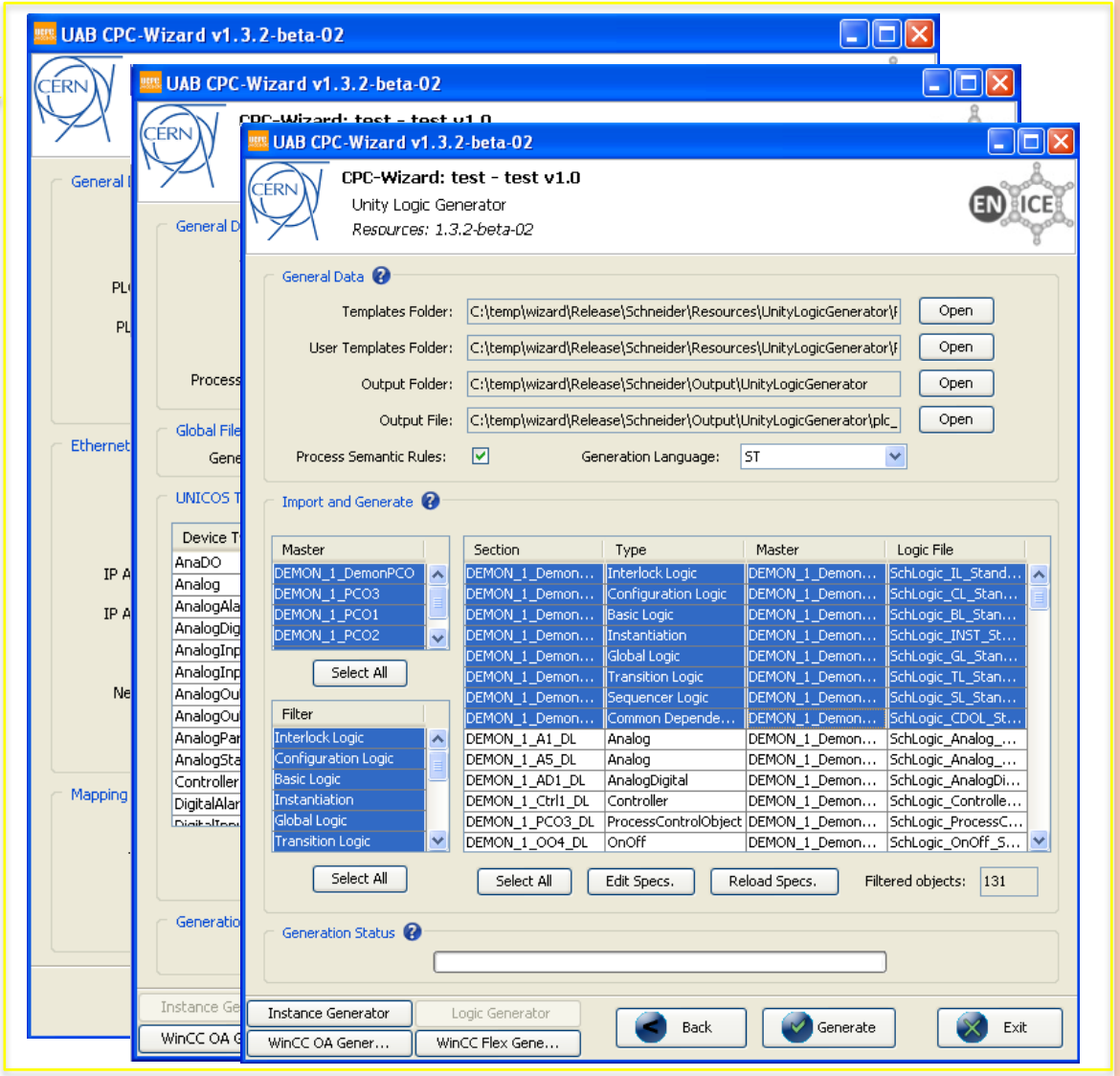
- Flexibility & Scalability
  - Improve Templates edition
  - Creation of new Objects
  - Introduce new platforms (e.g. Labview) and/or UNICOS components (e.g. CIET)
- Performance
  - Large size user application
- User friendly
  - Graphical interface based on a Wizard
- Versioning management

UAB: UNICOS Application Builder



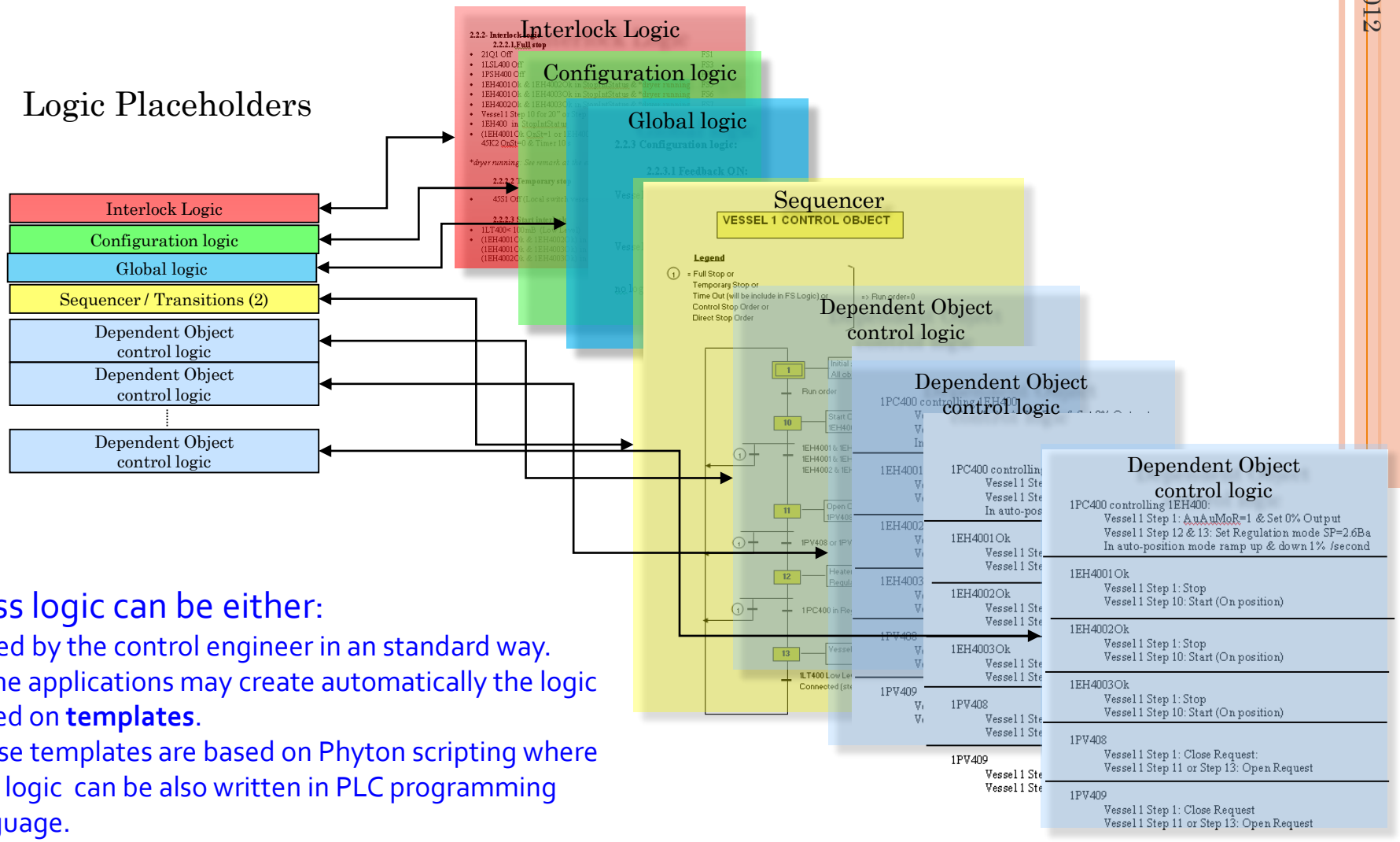
WORKFLOW BASED ON WIZARDS





UNICOS CPC LOGIC PLACEHOLDERS

For Each PCO the process engineers supply the logic associated to each PCO in a template document (WORD)



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 Phyton scripting where PLC logic can be also written in PLC programming language.

Control Logic Development

- Possibilities according to repeatability and developers habits/skills

1. Generation of Default Logic Code
   - User logic written by hand typically inside the PLC software
   - Need to maintain specs + PLC software up to date
   - Generation of logic is performed only once

2. Generation of User Logic Code
   - Creation of User Logic Templates and specs including logic parameters
   - Need to maintain specs + logic templates up to date
   - Need to generate logic at each logic modification

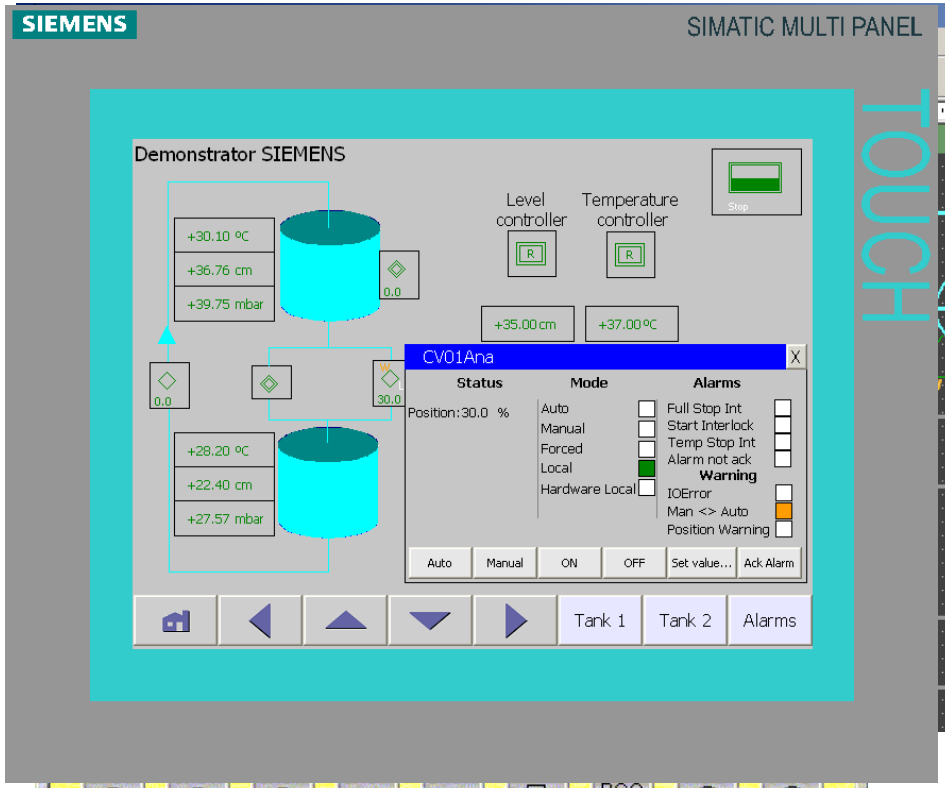
PCO CONTROL LOGIC

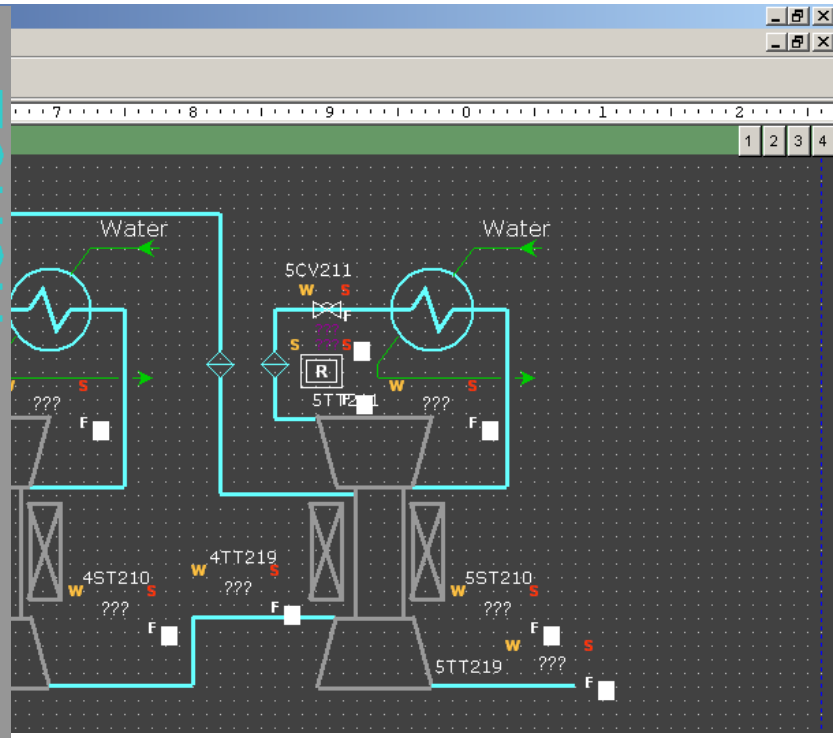
- Generated automatically:
  - **BL**: Basic Logic : Basic logical links between objects and PCO
  - **CDOL**: Common Dependent Object Logic: Auto request of all dependent objects (alarm ack. included)

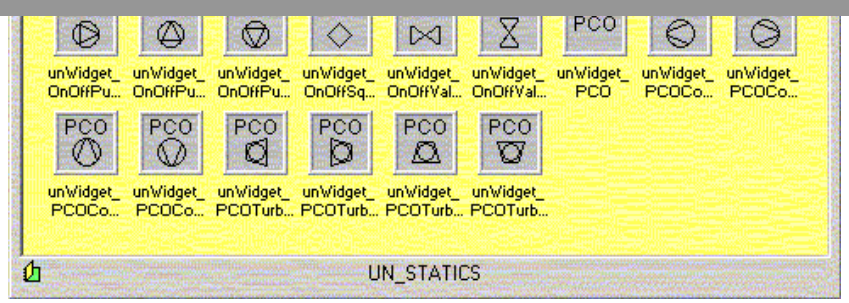
- To complete by user:
  - **CL**: Configuration Logic : State calculation of the PCO (On/Off)
  - **IL** : Interlock Logic : Specific logic for PCO interlocks
  - **DL**: Dependent Logic : Object Specific logic depending of the PCO
    - OnOff, Analog, Anadig, AnaDO, Controller and MFC
  - **TL**: Transition Logic : Compute transitions of a grafcet (*optional*)
  - **SL**: Sequencer Logic : grafcet/stepper (*optional*)

HMI SYNOPTICS

- Manual intervention (or automatic if known a priori)

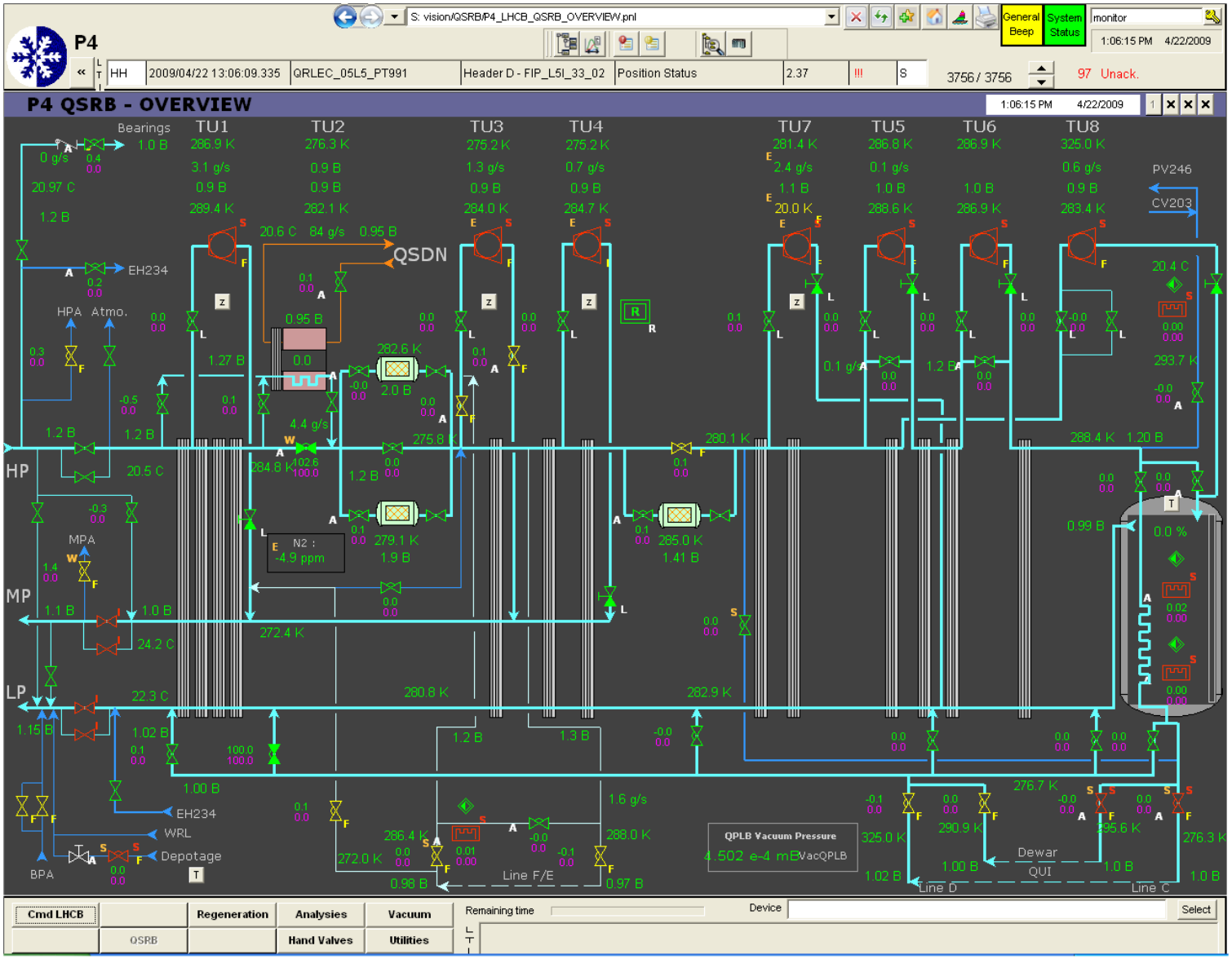






Synoptic design
- by drag & drop (manual operation)
- Automatically created (xml)

UNICOS CPC HMI



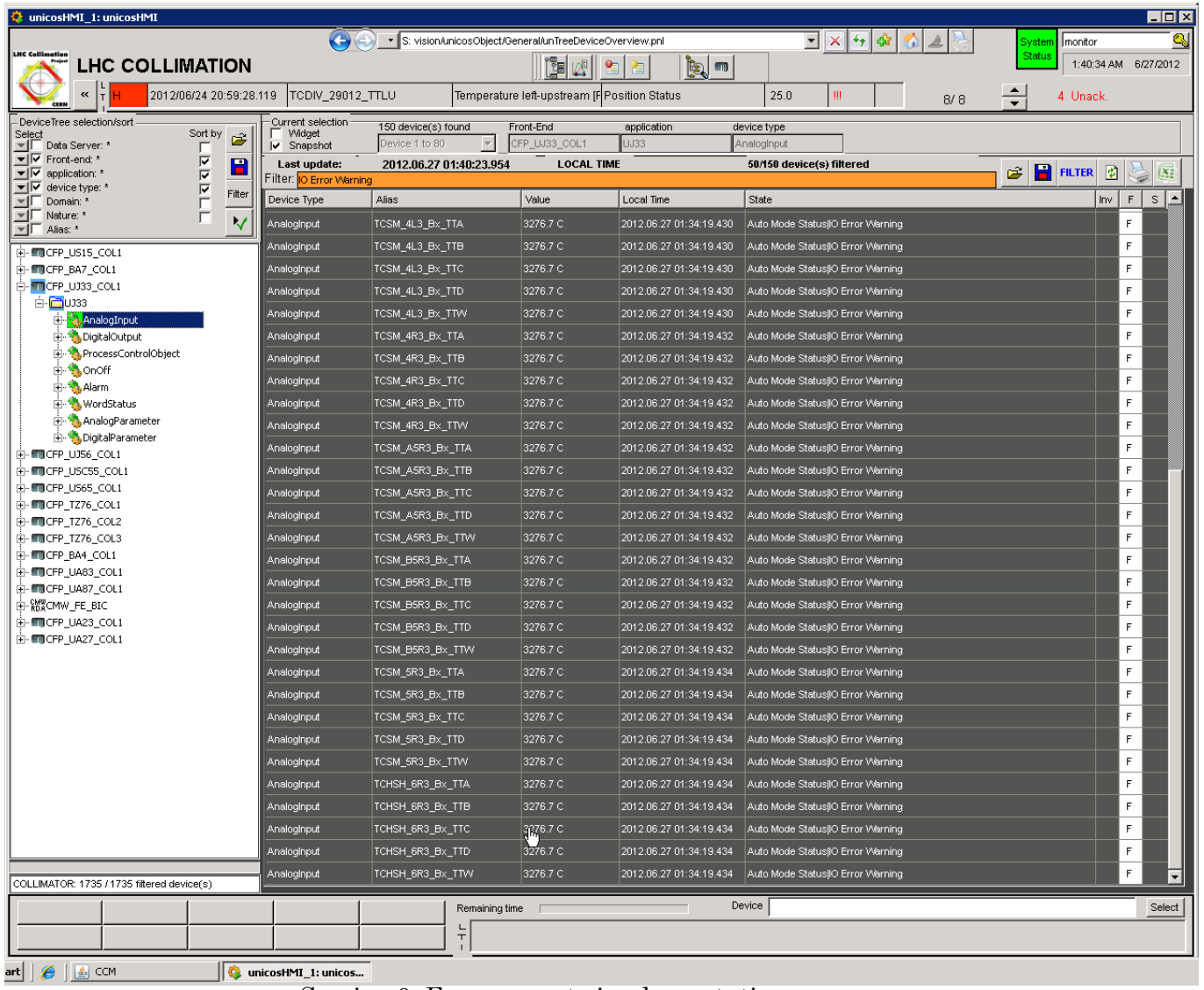
General Features: HMI Colors

- Reduced set of colors

	Faceplate			Widget		
INFORMATION/ORDERS/STATUS						
Status (status/modes)	Green {RGB: 0,255,0}	56.9			56.9	
Request (Request/Orders)	Green {RGB:0,175,0}	56.9			56.9	-
Parameters / Information	Black	4.56	√		-	-
PROCESS ALARMS						
Alarms	Red	-			34.4	
WARNINGS/FORCED						
Warnings	Orange				43.1	
Forced by operator	Yellow		-		45.5	
ABNORMAL SITUATION						
Invalid/Old data	Cyan	56.9				
Data not accessible	Violet	???				

COMMISSIONING

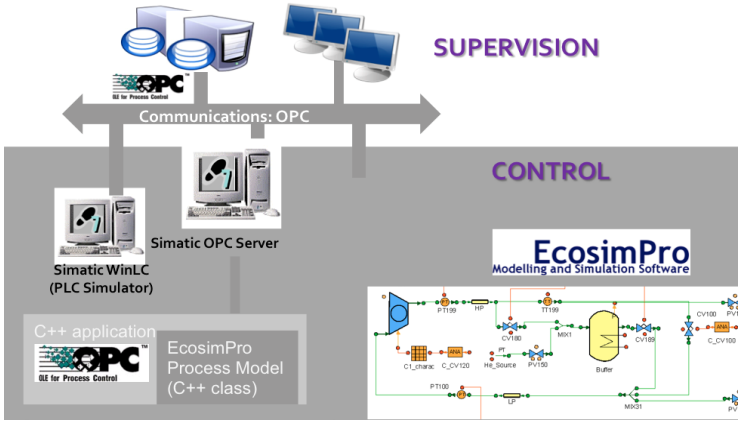
- Early I/O commissioning : Immediately after the I/O specs generation



FAT vs SAT

- For critical systems:
  - Factory Acceptance Test
  - Site Acceptance Test

by **Virtual Commissioning** (Modeling and Simulation)

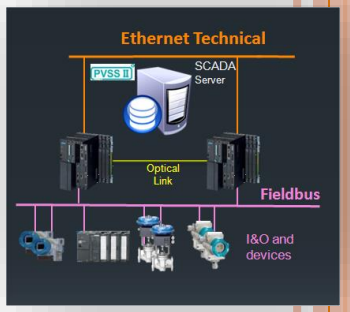


UNICOS CPC FEATURES

- **DEVELOPMENT**: Provides the developer with tools to produce rapid control applications . **Automatic generation of code**
- **COMMISSIONING**: Allows early plant commissioning, reducing check tasks to the instruments and electrical level (cabling). Access to devices without SCADA development.
- **OPERATION**:
  - Provides the operator a standard way to interact all the objects
  - Navigation capabilities between panels and trends: WWW browser-like, contextual buttons, pop-up navigation
  - Access Control
- **DIAGNOSTICS**: Tools to diagnose problems (process alarms, Infrastructure: control components,…)
- **MAINTENACE**: Allows optimized code evolution and maintenance by a reduced development team

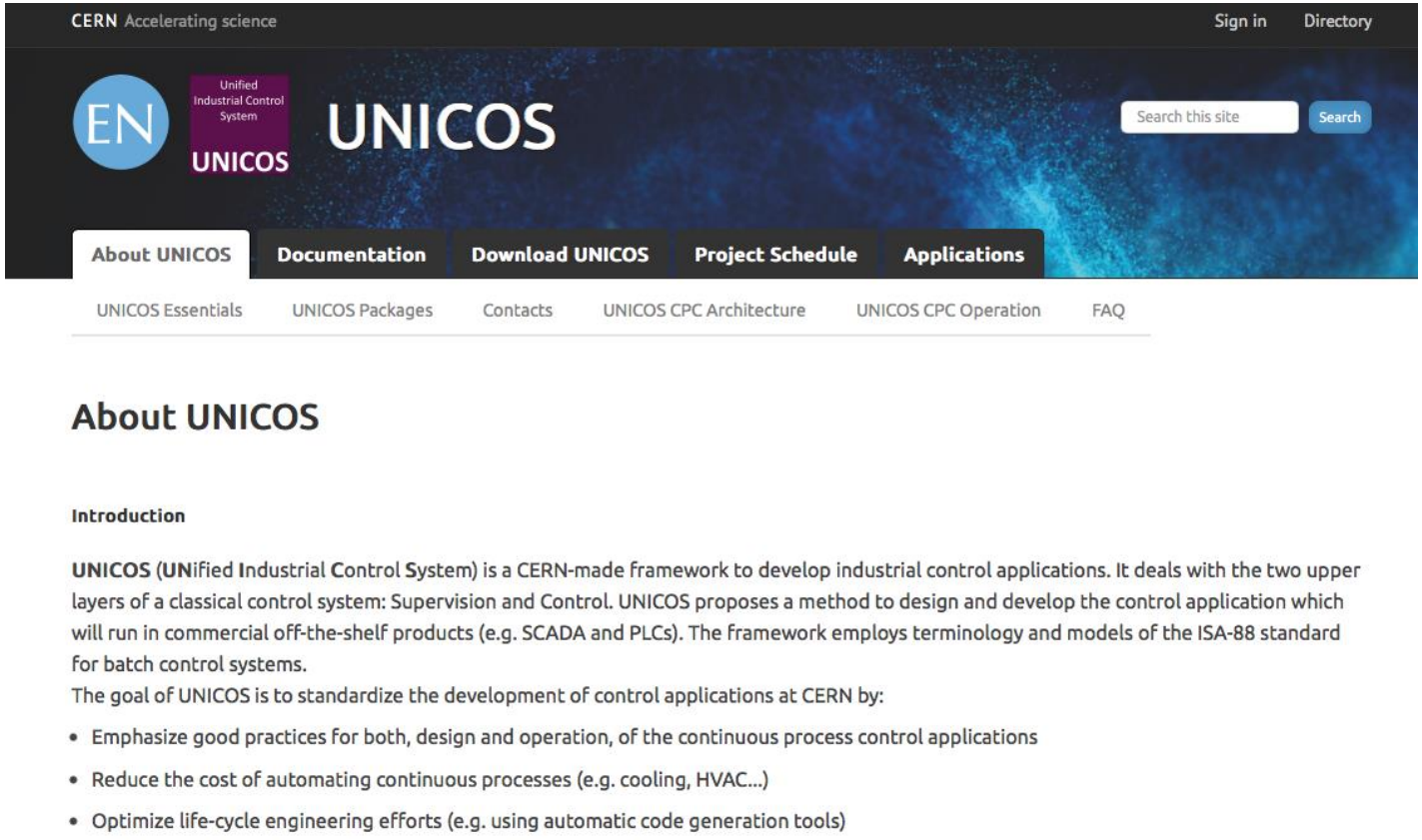
ONGOING WORKS

- New platforms
  - UNICOS with Labview as light supervisor for small size or lab projects
  - CoDeSys (Controller Development System)
    - Software tool which turns any PAC into an IEC 61131-3 programmable controller
- New PLC architectures
  - PLC Redundancy
    - Increasing availability for critical applications (Siemens & Schneider)
  - Safety Instrumented Systems (SIS)
    - Integration of the SIS into the UNICOS CPC framework
- Integration of tuning and control algorithms in UCPC
  - Advanced and modern control
    - Lowering advanced algorithms to the PLC level (e.g. predictive control…)
  - Online controller tuning
    - Tools allowing the operator to tune the PIDs online



More Information

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



QUESTIONS