

Development of Slow Control for the BELLE II CO₂ Cooling System

Michael Ritzert (Heidelberg University),
Thorsten Röder (MPI, TUM),
Bogdan Lobodzinski (MPI, DESY)

Belle II PXD Slow Control Meeting
MPI Munich 23.09.2014

Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**
-

Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**
-

SC of the IBelle cooling plant

Task distribution:

GUI CS-Studio (BELLE)



Operator Console GUI

EPICS db, IOC (BELLE)



EPICS system:
database, IOC

TCP/Modbus
protocol

Control Logic Level (CERN)



PLC Process Control:
Schneider TSX*

TCP/IP protocol

Industrial Level (CERN)

Fieldbus
network

Field interfaces

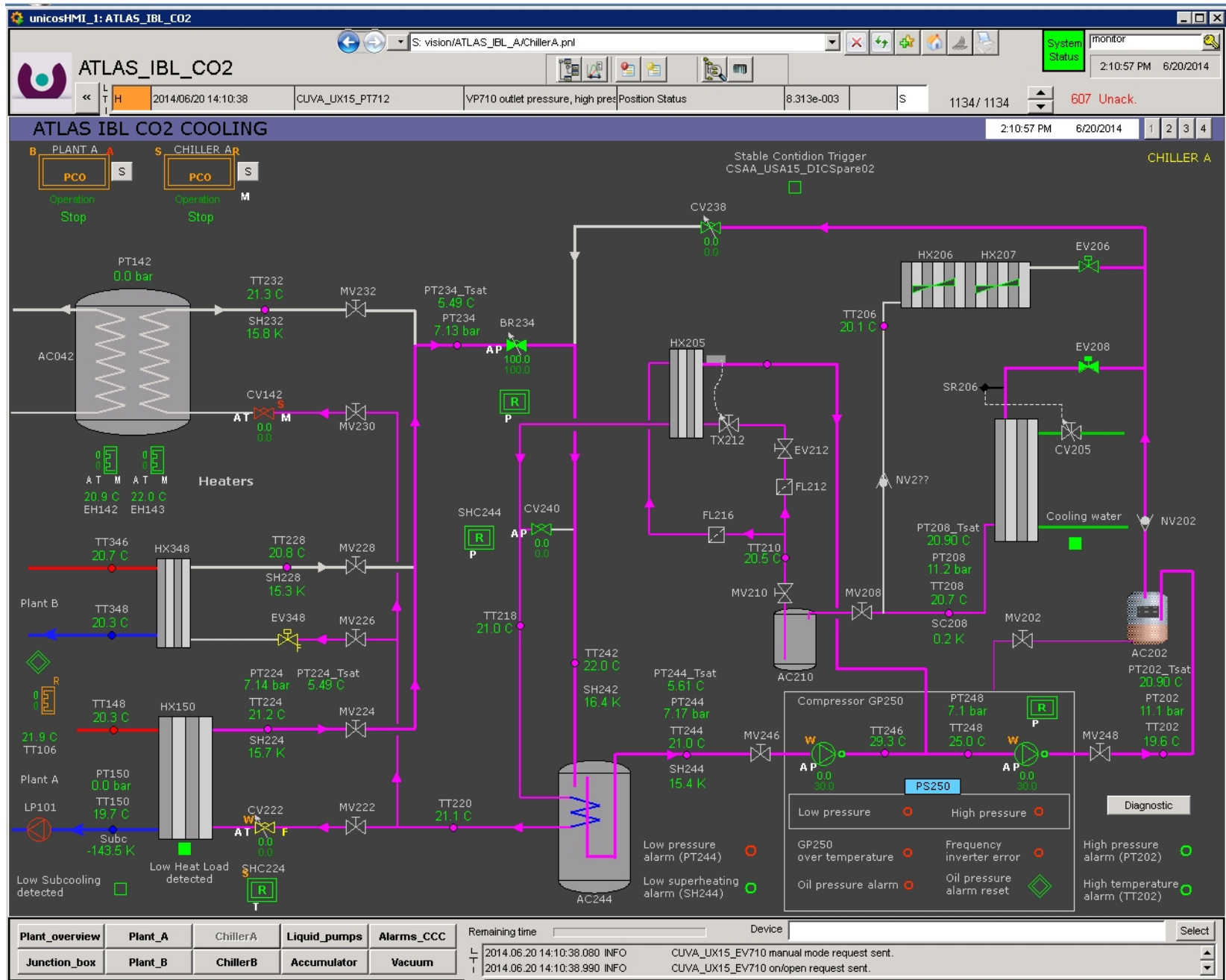


Cooling Plant

PLC = Programmable Logic Controller

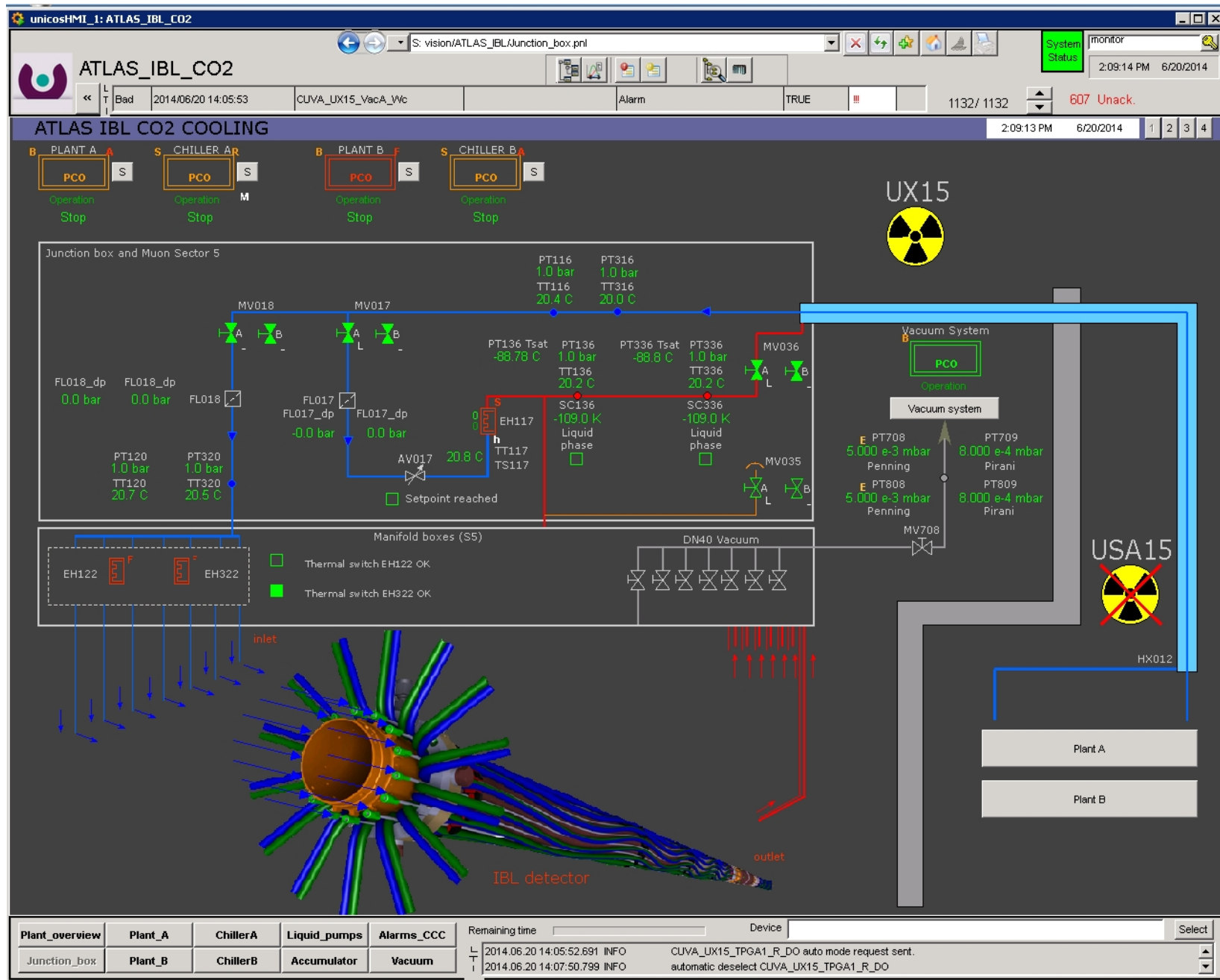
SC of the IBelle cooling plant

Our goal (from ATLAS IBL): Chiller A



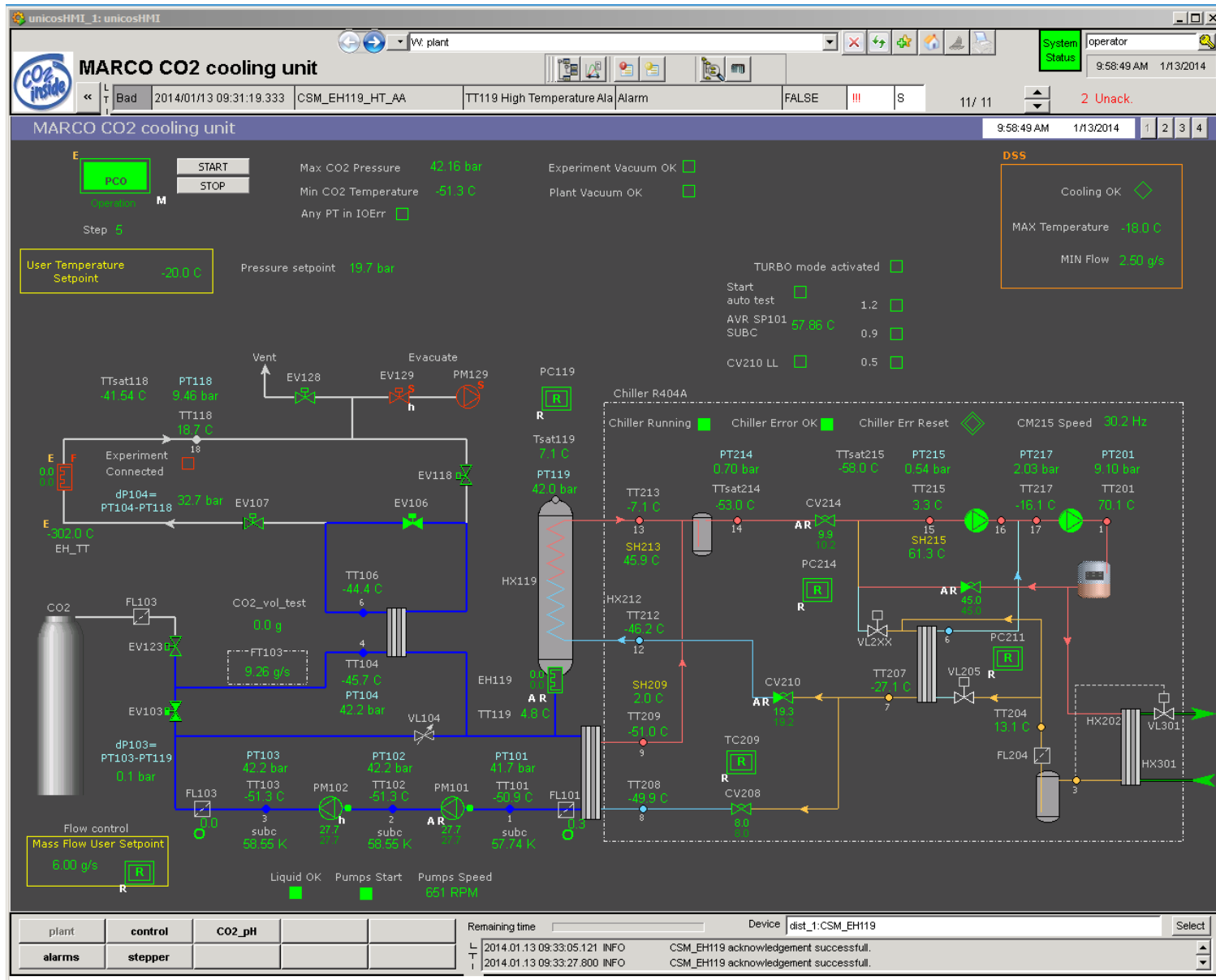
SC of the IBelle cooling plant

Our goal (from ATLAS IBL): Junction Box



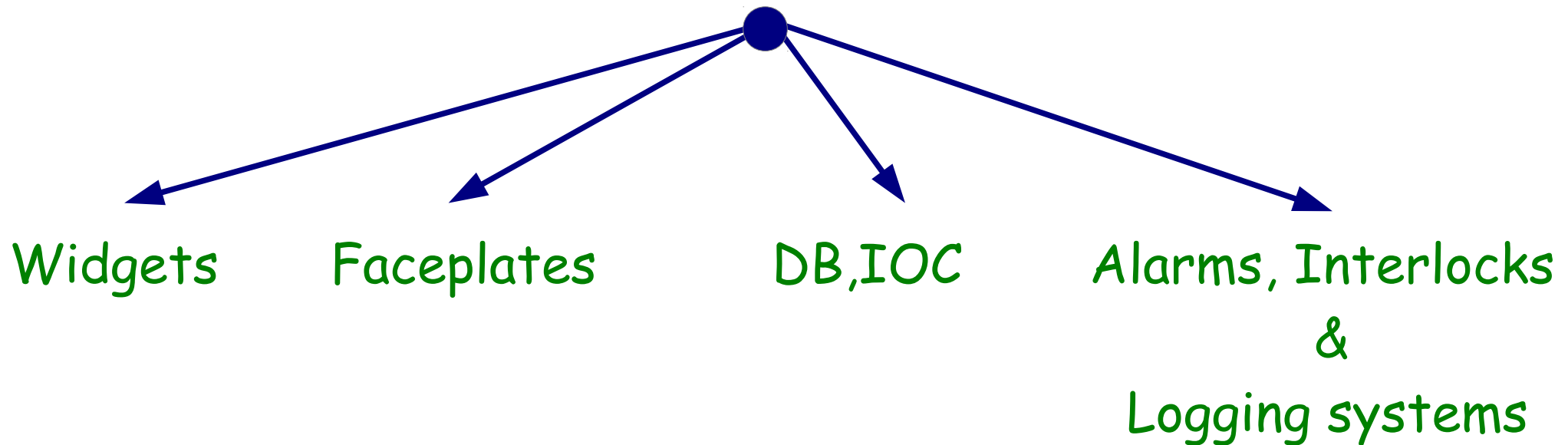
SC of the IBelle cooling plant

Another example: MARCO cooling system (plant scheme)



SC of the IBelle cooling plant

SC project development requires:



Widget = graphical representation of a device (valve, heater, sensor etc), process units and alarms

Faceplate = graphical front panel showing parameters of a selected device (a widget)

IOC = EPICS Input Output Controller for TCP/IP Modbus control protocol

SC of the IBelle cooling plant

Widgets represents a state of device like alarms, working modes, access to data etc

UNICOS-CPC Widgets animation

(W) DATA QUALITY (warnings)				
Object	Letter	Color	Priority	
Invalid data	all	N	Cyan	highest
Old (data not up to date)	all	O		
Field object with an alarm blocked IO object with IOError blocked	Field, I/O	B	Orange	
IO error	all excep	E		
IO simulated	xPAR, xSTATUS	S		
Forced <=> Auto	PCO,	W		
Manual <=> Auto	Field, I/O			
Position Warning				
Configuration Warning	AA	C	lowest	

(Ai) ALARM & INTERLOCKS				
Object	Letter	Color	Priority	
Full stop interlock	F	Red	highest	
Temporary Stop Interlock	S			
Start Interlock	I			
Alarm	A			
Position Alarm (Local)	Local	P		
Manual Restart Required (after a full stop interlock)	PCO, Field	R	Orange	lowest

(Ai) Info (Alarm objects)				
Object	Letter	Color	Priority	
Alarm condition	DA	A	Red	highest
High Threshold alarm	AA, AI,	HH		
Low Threshold alarm	AIR, AO,	LL		
High Threshold warning	AOR	H	Orange	lowest
Low Threshold warning		L		

(Ai) MASK & BLOCK info				
Object	Letter	Color	Priority	
Alarm Blocked (PLC)	PCO,	B	Yellow	highest
Alarm Masked (only SCADA)	Field,	M	Yellow	lowest
Event Masked (only SCADA)	all	e		

(M) Mode & Working State				
Object	Letter	Color	Priority	
Hardware Local Mode	HL	White	highest	
Local Mode	L	White		
Auto Mode	none			
Manual Mode	M	White		
Forced Mode	F	Yellow		
Inhibit Manual/Forced	h	White	lowest	

(X) Additional Information		
Letter	Color	
Controller mode (up to 3 letters)	A,M,F,L,R,P,T	White

(F) Feedback & (O) Order		Color
Feedback value		Green
Order value		Green

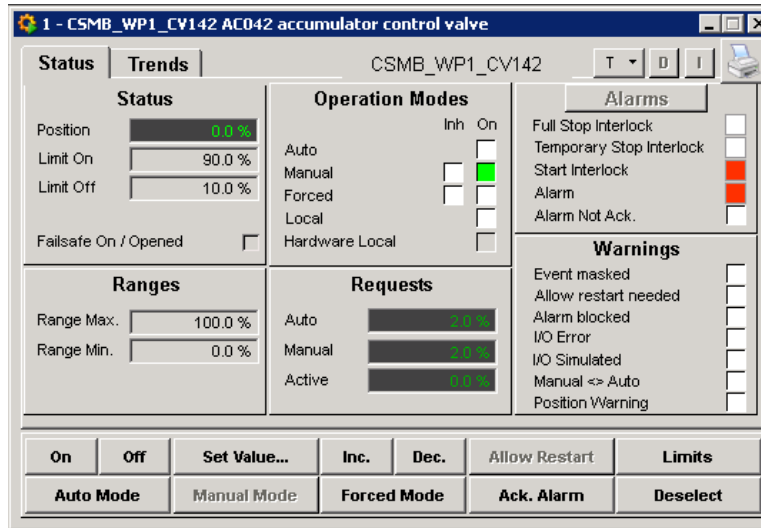
Analog	AI,AS	Analog	Analog	Analog	OnOff	AI,AS	PID	Analog	OnOff	OnOff	DA	AA	AA		
Data Invalid	No refresh Data	Data not connected	Forced Mode	Auto Regulated & order: 17.1 stat: 16.3	Warning & Forced Mode	Forced Mode & Event Masked	Manual & Regulation & Simulated	Inhibit Manual mode activate	Temporary Stop Interlock	Start Interlock active	Alarm On	Blocked alarm	Warning High		

picture from
the UNICOS
documentation:

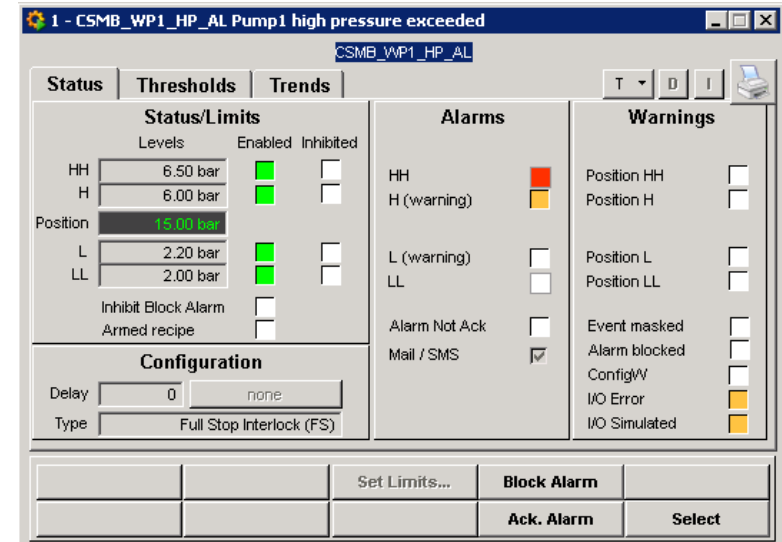
<https://j2eeps.cern.ch/wikis/display/EN/LHC-GCS+Help>

SC of the IBelle cooling plant

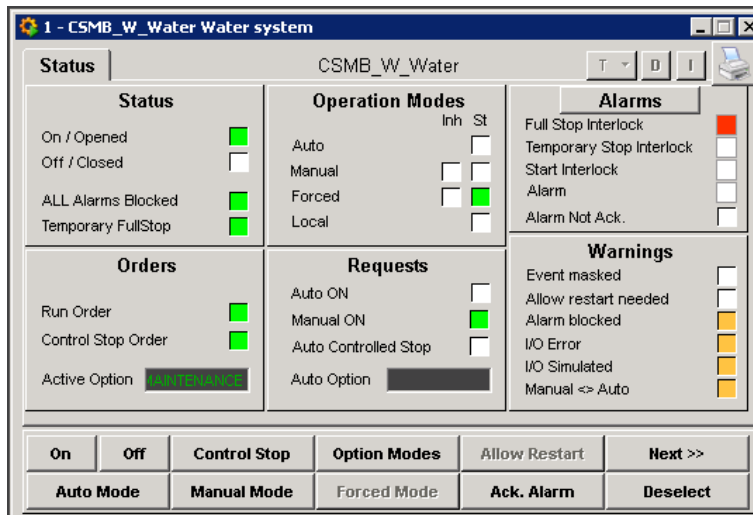
Faceplates shows parameters of a device (limits, actual values of data etc) or widget & allows to modify them



faceplate for Analog device



faceplate for Analog Alarm widget



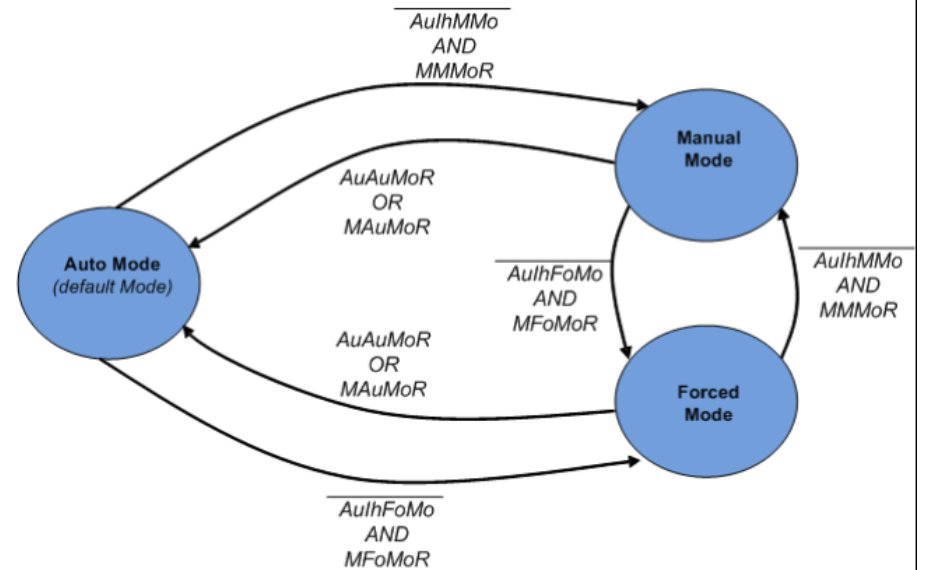
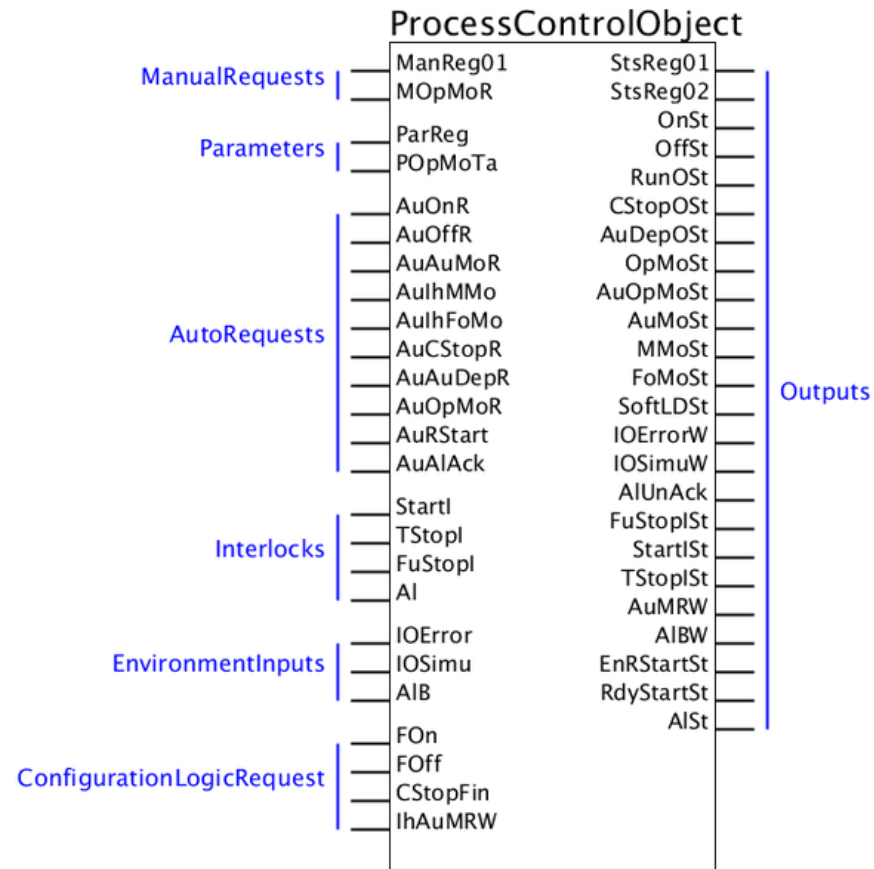
faceplate for Process Control Output widget

SC of the IBelle cooling plant

Faceplates

Details of signal description (left) & Mode manager (right)
(an example for Process Control Object)

Signal description



SC of the IBelle cooling plant

SC of the MARCO Cooling System based on the WinCC OA SCADA system (previous PVSS) is a set of objects:

Generated Objects:

#DigitalInput: 67
#AnalogInput: 34
#AnalogParameter: 48
#AnalogInputReal: 79
#ProcessControlObject: 3
#Controller: 15
#DigitalAlarm: 86
#AnalogAlarm: 60
#Analog: 8
#AnalogDigital: 10
#AnaDO: 4
#OnOff: 30
#AnalogStatus: 12
#DigitalOutput: 42
#AnalogOutput: 8
#AnalogOutputReal: 10

516 devices

For ATLAS IBL plant:

- 366 alarms and interlocks
- 81 faceplates

For IBelle we expect ~ 10 000 PVs

It is much too complex structure for manual development

SC of the IBelle cooling plant

CERN will prepare the IBelle cooling plant with SC based on the WinCC OA SCADA system.

Would it be possible to create the IOC and the DB for EPICS framework in parallel ?

The WinCC OA SC is created automatically by the system UNICOS-CPC (UNified Industrial COntrol System for Continuous Process Control)

BUT

UNICOS doesn't support EPICS

Integration with UNICOS

About UNICOS:

- a CERN-made framework to develop industrial control applications,
- UNICOS allows to design and develop the control applications which will run on a top of commercial SCADA systems (WinCC OA and PLCs (Siemens & Schneider)),
- well suited to large applications: UNICOS allows for
 - a unification of developed standards,
 - simplifies developing efforts

Current Status of UNICOS:

- based mainly on Java and on Python,
- available only for Windows OS and WinCC OA and LabView,

Planned extension:

- to add EPICS software environment. Such interest is expressed also by: ITER & ISS

Integration with UNICOS

SC project development model using UNICOS:

- 1) Preparation of the PLC hardware configuration
- 2) Preparation of the Excel specification: detailed list of all Data Points (PVs) with description, widget types, alarm messages, addresses of all inputs and outputs, parameters, aliases and dependencies (master, parents and children)
- 3) Preparation of the PLC Logic
- 4) Generation of the PLC instances & upload to the PLC
- 5) Generation of the WinCC OA objects

Integration with UNICOS

SC project development model using UNICOS:

- 1) Preparation of the Excel specification: detailed list of all Data Points (PVs) with description, widget types, alarm messages, addresses of all inputs and outputs, parameters, aliases and dependencies (master, parents and children)
- 2) Preparation of the PLC hardware configuration
- 3) Preparation of the PLC Logic
- 4) Generation of the PLC instances & upload to the PLC
- 5) Generation of the WinCC OA & EPICS (DB, IOC) objects

All necessary widgets, faceplates & general schemes have to be created separately.

Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**
-

Testbed setup

The project is realized in cooperation with CERN:
Lukasz Zwalinski, Maciej Ostrega, Enrique Blanco Vinuela

- For the development purpose we have at CERN working testbed setup with hardware assigned to the IB Belle cooling plant:

Type	Ref..
Electric extension rack (6 slots)	TSXRKY6EX
line terminators	TSXTLYEX
Power supply module	TSXPSY5500M
CPU module	TSXP575634M
SRAM memory card 1.7 MB SQD	TSXMRPC01M7

The system is not connected to any other sort of hardware.
We don't use Analog (Digital)/In Out cards.

- **EPICS: preparation of DB & IOC:**

- **IOC:**

- the TCP/IP Modbus driver is used without modifications,

- **DB:**

- PVs naming convention: it is save to adopt the naming introduced by CERN for D(ata) P(oints),

- an automatic creation of the DB is implemented into the UNICOS framework - is ready and working (MR),

- requires start of the UNICOS software on Windows 7 platform :(

EPICS DB & IOC

- **EPICS:**
automatic creation of
the DB:

Step 1

UCPC # epics_start_01 v1.0 # Wizard v1.6.0

CERN

CPC Wizard: EPICS_project_1 - epics_start_01 v1.0
SCHNEIDER PLC Specifications
Resources: 1.6.0

EN ICE

General Data ?

PLC Name: CFP_PLC1

PLC Type: PREMIUM

Recipes ?

Enable recipes:

Max. number of recipe values: 1000

Recipe Buffer Starting Address: 40000

Activation Timeout (s): 100

Ethernet Parameters ?

PLC IP Address: 1.1.1.1

PLC Gateway: 1.1.1.4

PLC Network Mask: 255.255.0.0

DS1 IP Address: 1.1.1.2

DS2 IP Address: 1.1.1.3

MODBUS Parameters ?

MODBUS Unit Address: 25

XWay Network: 1

XWay Station DS1: 106

XWay Station DS2: 107

Binary Tables per Cycle: 2

Analog Tables per Cycle: 2

Mapping ?

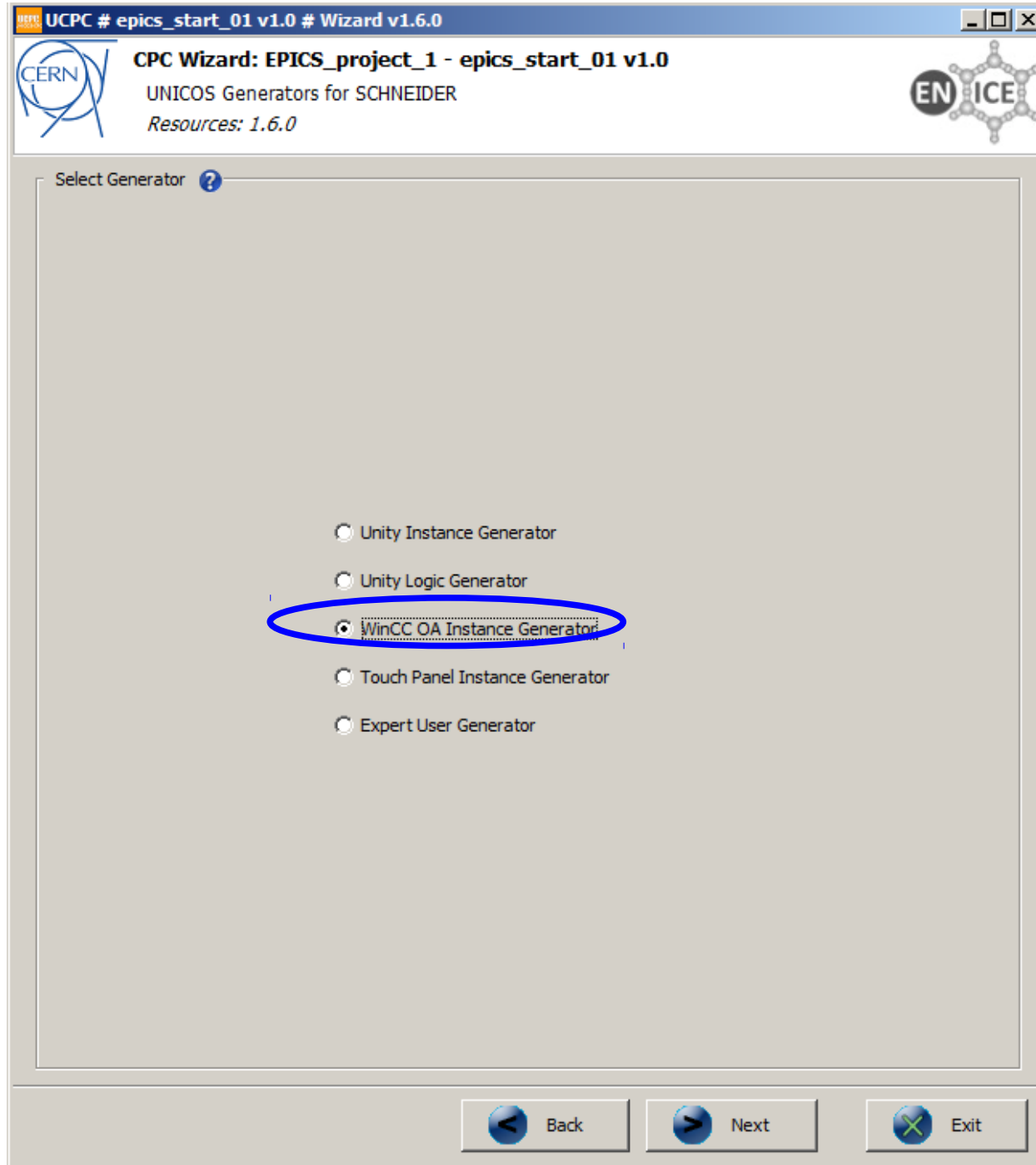
Transmit Buffer Starting Address: 1000

Receive Buffer Starting Address: 20000

Back Next Exit

- EPICS:
automatic creation of
the DB:

Step 2



EPICS DB & IOC

- EPICS:
automatic creation of
the DB:

Step 3

UCPC # epics_start_01 v1.0 # Wizard v1.6.0

CERN

CPC Wizard: EPICS_project_1 - epics_start_01 v1.0
WinCC OA Instance Generator
Resources: 1.6.0

EN ICE

General Data ?

Templates Folder: C:\Users\bogdan\Desktop\UNICOS\UNICOS_SETUP\Resources\WinCCOAInstanc

Output Folder: C:\Users\bogdan\Desktop\UNICOS\UNICOS_SETUP\Output\WinCCOAInstanceGe

DB File: C:\Users\bogdan\Desktop\UNICOS\UNICOS_SETUP\Output\WinCCOAInstanceGe

Post Process User Template: Browse Clear C:\Users\bogdan\Desktop\UNICOS\UNICOS_SETUP

Process Semantic Rules: Post Process User Template:

UNICOS Types To Generate ?

Device Type	Instances
AnalogDigital	2
AnalogInput	2
AnalogInputReal	2
AnalogOutput	1
AnalogOutputReal	2
AnalogParameter	5
AnalogStatus	2
Controller	4
DigitalAlarm	3
DigitalInput	4
DigitalOutput	2
DigitalParameter	2
Local	2
MassFlowController	2
OnOff	2
ProcessControlObject	2
WordParameter	2

Select All Edit Specs. Reload Specs.

Generation Status ?

Instance Logic Expert

WinCC OA Touch Pa...

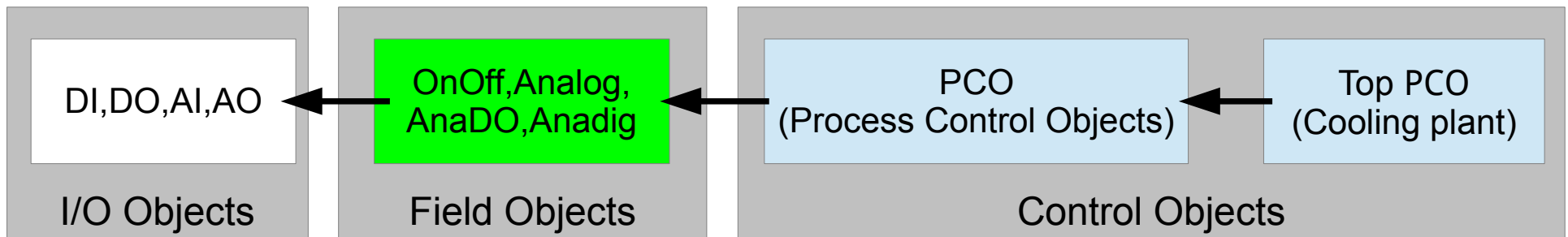
Back Generate Exit

Available objects

Widgets

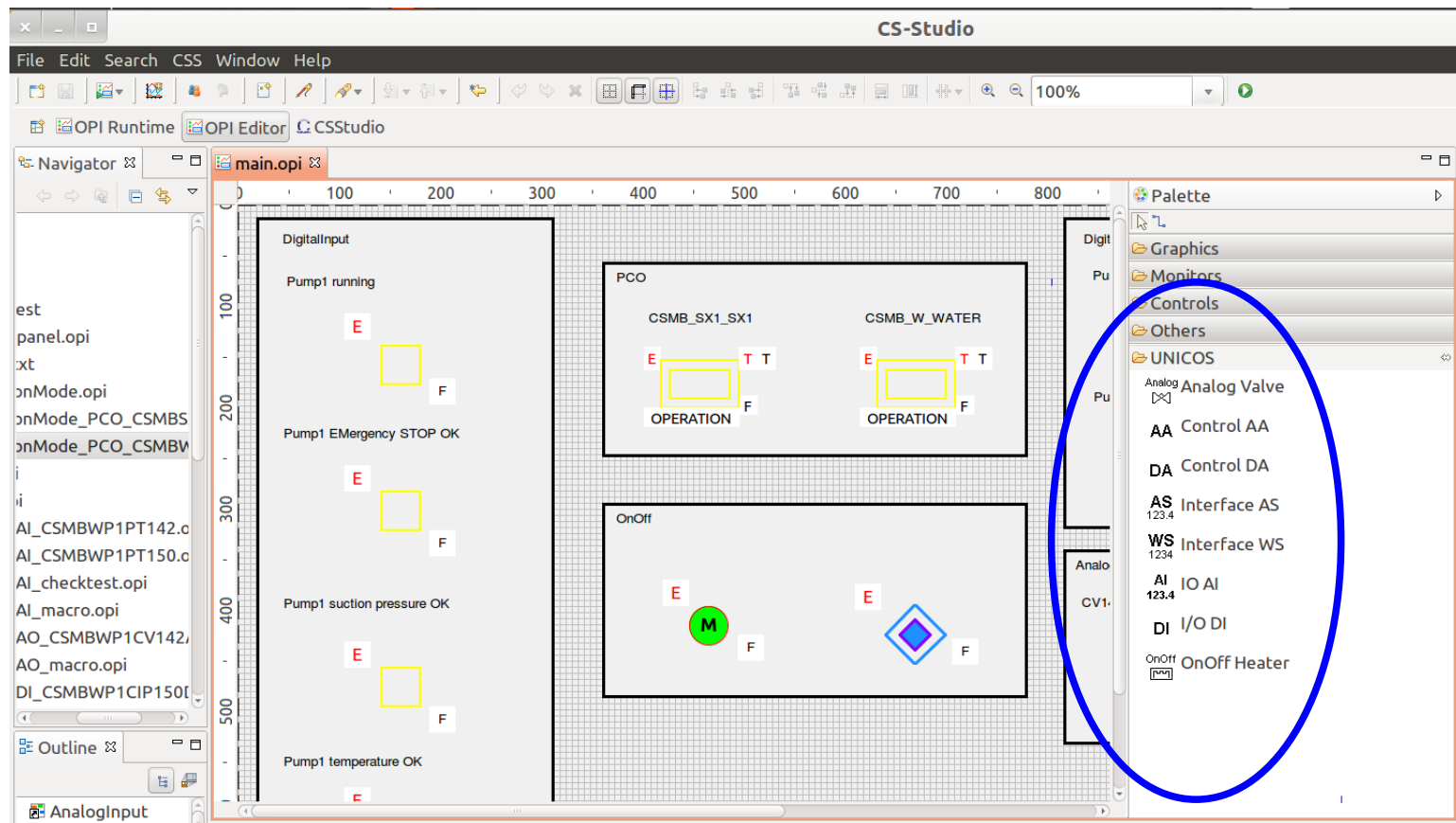
- **I/O widgets:**
 - Analog Input (e.g. temperature transmitter)
 - Digital Input (e.g. end contact)
- **Field widgets:**
 - OnOff objects (e.g. on/off valve)
 - Analog objects (e.g. control valve)
- **Control widgets:**
 - Digital alarm objects (e.g. valve access)
 - Analog alarm objects (e.g. pressure too high)
- **Interface widgets:**
 - Word/Analog Parameters: send a signal from EPICS to PLC (e.g. threshold)
 - Word/Analog Status: send a signal from PLC to EPICS (e.g. stepper position)

The specification contains only widgets available as CSS objects.



Available objects

Widgets



The set of widgets is a part of the CSS release prepared for BELLE II
Usually available on: <https://sussrv01.ziti.uni-heidelberg.de/~ritzert/>

Faceplates

Nearly final version of all faceplates is available on a website:
<https://belle2.cc.kek.jp/~twiki/bin/viewauth/Detector/PXD/UNICOS:>

Things to do

Widgets

- I/O widgets:
 - Analog Output (e.g. control valve position order)
 - Digital Output (e.g. onoff valve position order)
- Field widgets:
 - Analog objects (e.g. valves controlled by on/off pulses)
 - AnaDO objects: OnOff + Analog (motor, heater)
 - Local Objects (e.g. manual valve)
- Control Objects:
 - Controller Objects
 - Process Control Objects: PCO (e.g. compressor station)
- Interface widgets:
 - Digital Parameters: send a signal from EPICS to PLC (e.g. threshold)
 - Digital Status: send a signal from PLC to EPICS (e.g. stepper position)
-

Things to do

Another subjects

- **Access Rights to the CSS control panel of the IB Belle plant:**
 - The issue have to be evaluated.
 - A potential solution which could be taken into account as a concept for the EPICS-CSS is the **Role-Based Access Control (RBAC)** (not used at CERN).
- **Alarm & logging systems:**
 - Will be integrated with the SCs of another part of the detector. Should not be too different than other "Alarm" parts of the detector.
 - Should contain Alarm handler, Archiver & access to the Message log system
 - Not yet fully defined
- **Tests with real test cooling system**

Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**
-

Proof-of-concept demonstration

Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**
-

Summary

- Usage of the EPICS-CSS is possible as a base for the SC of the CO₂ cooling plant
 - The alarm & logging systems are one of the most complex tasks in the project
- Full unification of the EPICS & the CSS with the UNICOS CPC framework requires a separate project
- Before the final implementation the SC have to be checked with a real test cooling system

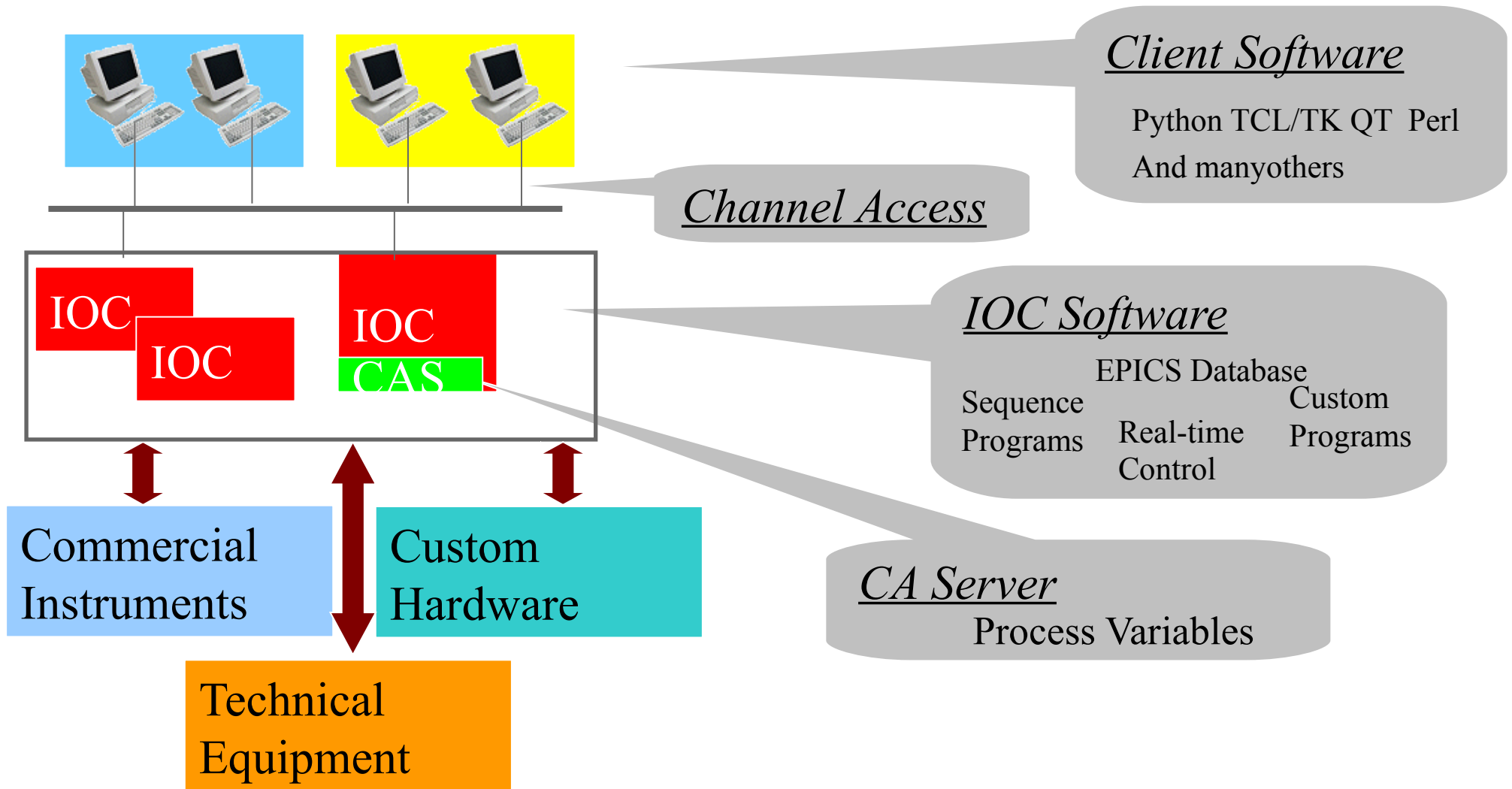
Agenda

- **Introduction**
 - S(low) C(ontrol) of the IBBelle cooling plant
 - Integration with UNICOS-CPC
- **Status of the SC development**
 - EPICS DB & IOC
 - Available objects
 - Things to do
- **Demo**
- **Summary**
- **Backup**

-

EPICS overview

- EPICS SCADA system - well suited to slow controls ~50 ms +



CA → Channel Access; IOC → Input Output Controller;
PV → Process Variable; OPI → Operator Interface (client);

Some features of EPICS :

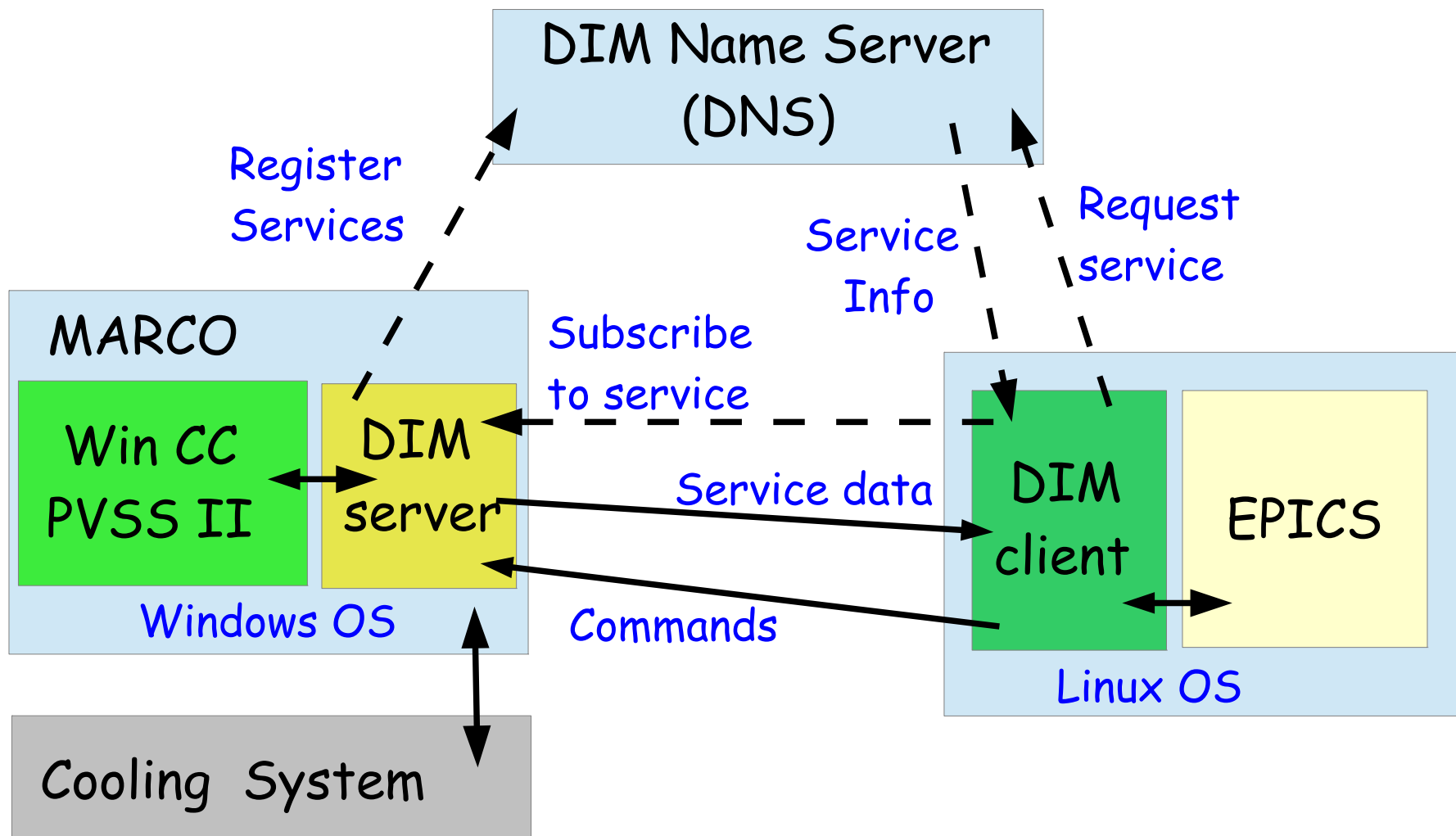
- distributed IOCs, edge driven, modular & heterogeneous,
- data & process flows can be linked into a single record,
- supports:
 - I/O to/from hardware (industrial standards), conversion, alarms, monitor, links

Operator Interface Tool (GUI) :

- several tools are supported:
 - MEDM (Motif-based Display Manager),
 - EPICS Qt (EPICS CA widgets for Qt GUI),
 - **CSS (Control System Studio)**
 - > based on Java/Eclipse IDE (Integrated Development Environment),

Monitoring of the MARCO cooling system

PVSS - EPICS interface based on the DIM
(Distributed Information Manager)



Monitoring of the MARCO cooling system

Monitored points of the MARCO cooling system:

- Temperature After Detector
- Detector
- CO2 flow
- Temperature Before Detector

