UNICOS: UNIFIED INDUSTRIAL CONTROL SYSTEM

CPC (CONTINUOUS PROCESS CONTROL)

BASIC COURSE

SESSION 0:

UCPC FROM SPECS TO IMPLEMENTATION

CERN, UNICOS team
UNICOS AND CPC

- **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 AND CPC

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

JCOP Framework

WinCC OA (PVSS)

UNICOS

PLCs CPC

UCPC package

Cryogenics
Gas
Collimators Interlocks
Cooling and Ventilation

Process Control applications
A LOOK TO THE PAST

• [1998] UNICOS (UNified Industrial Control System) 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

Session 0: From specs to implementation
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

Session 0: From specs to implementation
APPLICATIONS

- 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 ESSENTIALS

- 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

Session 0: From specs to implementation
UNICOS CPC OBJECT MODEL

Object MODEL

- Process inputs
- Manual Requests
- Auto. Requests
- Parameters
- Orders
- Status

PLC internal Object Logic

Objects status

- Plant Operator
- Parent Object

Process output or Child Auto Requests

Information to other object or to operator

Control Engineer

Session 0: From specs to implementation
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**

Session 0: From specs to implementation
UNICOS-CPC OBJECTS

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

Session 0: From specs to implementation
## 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
- FailSafe Pos
- Req Local Oper
- Dev oper locally
- Manual mode
- Motion deviation
- Stop r (~open)
- Stop r (~closed)
- Compressed air
- Cmd open sent
- Cmd close sent
- Param change
- Simulation
- N/A
- Ctrl Fault
- Control inactive
- Self test
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)

Control Devices

Field Devices

I/O Devices

Interface: I/O Boards-Fieldbus-Other PLCs

Session 0: From specs to implementation
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.

Session 0: From specs to implementation
DO OBJECT CONNECTIVITY

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

Logical connectivity
Physical connectivity

Process Control Devices

Field Devices

I/O Devices

I/O Boards

Session 0: From specs to implementation
Status of AI Objects may be used by all objects anywhere in the program.
AO Objects receive their orders only from Field Device Analog.

AO OBJECT CONNECTIVITY

Session 0: From specs to implementation
Status of Local Objects may be used by all objects anywhere in the program.

Session 0: From specs to implementation
ONOFF OBJECT CONNECTIVITY

Session 0: From specs to implementation
ANALOG OBJECT CONNECTIVITY (no PID)

Session 0: From specs to implementation
ANADIG OBJECT CONNECTIVITY (NO PID)

Session 0: From specs to implementation
Session 0: From specs to implementation
UNICOS CPC
ENGINEERING LIFE CYCLE

Decomposition  Specifications  Automatic Code Generation  Logic specifics & Synoptics  Deployment  Reverse Engineering

Session 0: From specs to implementation
DECOMPOSITION METHOD

Session 0: From specs to implementation
SPECIFICATIONS

UNICOS CPC Specs (xls/xml file)

Functional Analysis + Logic specification (Word templates)

Session 0: From specs to implementation
Session 0: From specs to implementation
Workflow based on wizards

Session 0: From specs to implementation
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.
UNICOS CPC HMI

Session 0: From specs to implementation
HMI SYNOPTICS

- Manual intervention (or automatic if known a priori)

---

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

- Check out the web page:
  - [http://www.cern.ch/unicos](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)