UNICOS EVOLUTION: CPC VERSION 6

E. Blanco Viñuela, J.M. Beckers, B. Bradu, Ph. Durand, B. Fernandez Adiego, S. Izquierdo Rosas, A. Merezhin, J. Ortola Vidal, J. Rochez, D. Willeman (CERN, Geneva, Switzerland)

CERN
icalepcs 2011

ABSTRACT

The UNICOS (UNified Industrial COntrol System) process control package has been reformulated as the **UNICOS CPC (Continuous Process Control) package** and a **reengineering process** has been followed. The drive behind this procedure was (1) being able to upgrade to the new more performing IT technologies in the automatic code generation, (2) being flexible enough to create new aditional device type types to cope with other needs (e.g. Vacuum or Cooling and Ventilation applications) without major impact on the framework or the PLC code baselines and (3) enhance the framework with new functionalities.

UNICOS Model

- The creation of a UNICOS model allows a proper homogenization of the different device types (objects) and also a sound base to develop new objects. The model is supported by a meta-model describing the properties of the model.
- **TCT** (**Types Creation Tool**) provides a drag&drop based mechanism to build new device type definition described in XML (eXtended Markup Language).

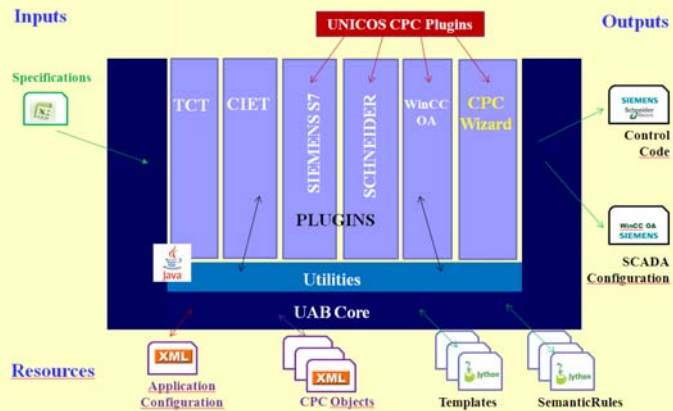UNICOS Device Types (Categories)

UNICOS model

UNICOS Device Types (Objects) & Control Specifications
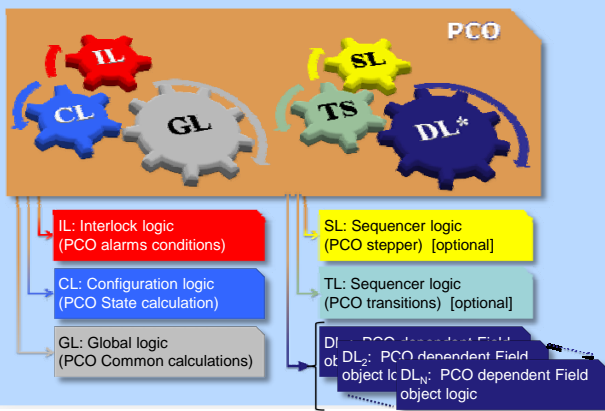
Code Generation Tools

Control engineers require appropriate flexible and scalable factory automation tools to develop control systems. UAB (UNICOS Application Builder) allows them to automatically generate such applications.

Inputs — Specifications — UNICOS CPC Plugins — Outputs

TCT | CIET | SIEMENS S7 | SCHNEIDER | WinCC OA | CPC Wizard

PLUGINS

Utilities

UAB Core

Resources — Application Configuration — CPC Objects — Templates — SemanticRules

Control Code — SIEMENS

WinCC OA SIEMENS

SCADA Configuration

## UNICOS CPC v6 Enhancements

Process Control

- The SIEMENS S7 CPC control baseline has been redesigned to allow new device types being added without modifying the PLC application architecture.
- The **control logic** is created in *Jython* using **standard logic Templates** which are associated to the PCOs (Process Control Units) and contains a predefined functionality (e.g. Interlocks, sequence transitions,...).

PCO

IL — CL — GL — SL — TS — DL*

IL: Interlock logic (PCO alarms conditions)

CL: Configuration logic (PCO State calculation)

GL: Global logic (PCO Common calculations)

SL: Sequencer logic (PCO stepper) [optional]

TL: Sequencer logic (PCO transitions) [optional]

DL₁: PCO dependent Field object logic
DL₂: PCO dependent Field object logic
DLₙ: PCO dependent Field object logic

Plant Operation

- The framework includes industrial local touch panels
- Dynamic recipes allows plant operators adapting the parameterization at runtime
- CERN HMI Color standardization
- Alarms level classification added
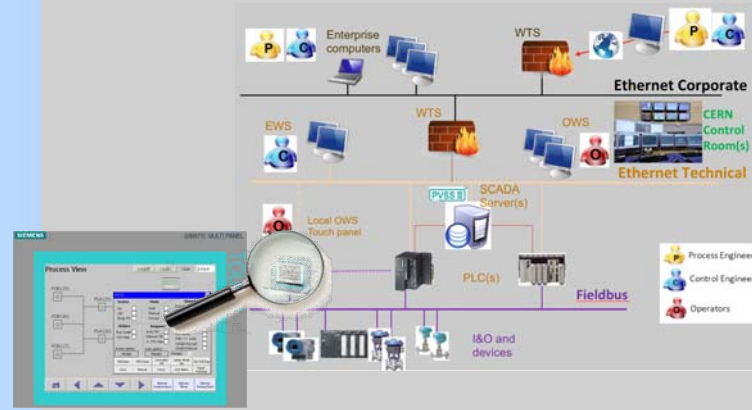
CONCLUSIONS

The UAB CPC6 package has been reengineered to add several improvements for developers and users (control system engineers and plant operators).
The developer, using UAB and its user-friendly CPC wizard, will obtain more flexibility and performance, allowing to add new CPC objects easily, new plug-ins for new platforms or semantic check rules. UAB also offers a set of services like a powerful version management which guaranties the maintenance of the control applications.

Plant operation is also improved by means of better adapted HMI coloring and alarms classification together with the option of operate the installations locally using industrial touch panels.

Redundancy — Regulation (Auto tuning) — UNICOS CPC — Safety — Advanced Control

FUTURE TRENDS

- Regulation: e.g. Controllers Auto-Tuning
- Advanced Control for complex processes
- Availability through redundancy
- Safety instruments systems

CERN — **Industrial Controls** **Engineering Department** — EN ICE