

UNICOS FRAMEWORK AND EPICS: A POSSIBLE INTEGRATION

M. Ritzert*, Heidelberg University, Mannheim, Germany,
 E. Blanco Vinuela, M. Ostrega, L. Zwalinski, CERN, Geneva, Switzerland

Abstract

UNICOS (UNified Industrial Control System) is a CERN-made framework to develop industrial control applications. It follows a methodology based on ISA-88 and provides components in two layers of a control system: control and supervision. The control logic is running in the first layer, in a PLC (Programmable Logic Controller), and, in the second layer, a SCADA (Supervisory Control and Data Acquisition) system is used to interface with the operators and numerous other features (e.g. alarms, archiving, etc.). UNICOS supports SIEMENS WinCC OA as the SCADA system. In this paper, we propose to use EPICS (Experimental Physics and Industrial Control System) [1] as the supervision component of the UNICOS framework. The use case is the control system of a CO₂ cooling plant developed at CERN following the UNICOS methodology, which had to be integrated in a control system based on EPICS. The paper describes the methods and actions taken to make this integration feasible, including automatic EPICS database generation, PLC communications, visualization widgets, faceplates and synoptics and their integration into CSS [2] and EPICS, as well as the integration with the BEAST alarm system.

INTRODUCTION

UNICOS (UNified Industrial Control System) is a CERN-made framework to develop industrial control applications. The package Continuous Process Control (CPC) provides a specific environment for process 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. Supervisory Control and Data Acquisition systems (SCADA) and Programmable Logic Controllers (PLC). The framework employs terminology and models of the ISA-88 standard for batch control systems that is also widely employed in continuous process control.

UNICOS-CPC is based on a well-defined set of standard device types covering most of the physical equipment (e.g. motors, valves, sensors, etc.) and the needs of a continuous process in terms of control; also it enforces the structure and provides a method for programming the process control logic.

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

- emphasizing good practices for both, design and operation, of the continuous process control applications

* michael.ritzert@ziti.uni-heidelberg.de

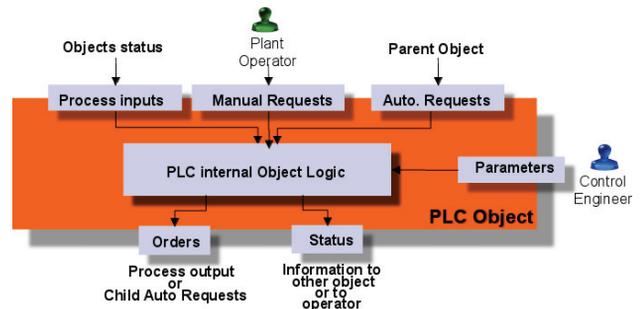


Figure 1: UNICOS object model.

- reducing the cost of automating continuous processes (e.g. cooling, HVAC, etc.) and
- optimizing life-cycle engineering efforts (e.g. using automatic code generation tools).

At CERN, UNICOS is used with WinCC OA as the supervision layer. This paper describes the effort to replace WinCC OA with an EPICS-based system, also employing Control System Studio (CSS) as the operator interface, the BEAST alarm system, and the Archiver Appliance to record PV history.

UNICOS CONCEPTS

UNICOS is an object oriented framework consisting of a package of programming tools facilitating control system development. It includes the baseline library (with a modular PID algorithm), the code generator with synoptic rules checker significantly reducing time of program development by a priority error identification, the skeleton templates, and examples of objects lists. The object definitions provided by UNICOS are split into:

- I/O Objects: Values to and from the hardware. E.g. Digital Input, Digital Output, Analog Input
- Field Objects: Physical equipment. E.g. OnOff, Analog, AnaDig, Controller, etc..
- Control Objects: Logic processing. E.g. PCO, Analog Alarm, Digital Alarm.

A typical UNICOS object model is presented in Fig. 1.

The UNICOS engineering life cycle presented in Fig. 2 is based on two initial documents: the functional analysis that is a Word document and an Excel XML specification file containing all object definitions, parametrization and hierarchy. The specification file is the primary input to the automatic code generation tool. This generation tool is equipped with a set of framework templates and user dependent logic templates to provide structured importation files for both PLC and SCADA, guaranteeing correct mapping between them.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

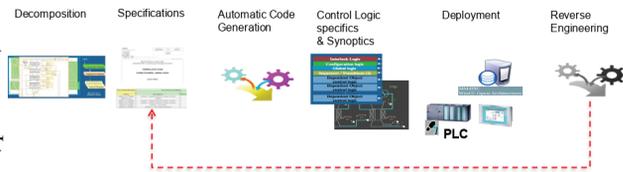


Figure 2: UNICOS design cycle.

METHODS

Overview

The port only targets the supervision layer of the control system. The actual slow-control system runs completely independently on the PLC's CPU. No changes to this part are required when moving between supervision by WinCC OA or EPICS. There are two big parts in the implemented porting procedure:

First, the UNICOS generator program has been modified to create its output in a format suitable for this port instead of the default WinCC OA format. A set of Jython modules used by the generator had to be modified. The output consists of several files.

The generated EPICS database contains the mapping from PLC register name (also the PV name) to PLC register address, and all PV metadata that can be extracted from the UNICOS project definition and put into the EPICS PV, such as the value's unit, display limits, display precision, etc.. Alarm limits are also put into the PV but are used mainly for display purposes. The actual alarm management is performed fully inside the PLC.

Also written out is the definition of the relationships between the various UNICOS objects in the system. Possible relationships are e.g. parent/child, or between an alarm object and an object put into interlocked state on alarm activation.

After the generator has written out the system definition, a set of Python scripts is used to create template OPIs containing widgets for all of the system's objects, and also to transform existing WinCC OA displays into CSS OPIs, if desired.

Interface to the PLC

For Schneider PLCs, the standard modbus protocol is used to interface the PLC with the SCADA system. The EPICS modbus driver [3] could be used without modifications to establish the communication between EPICS and the PLC.

In order to generate the required mapping from register name to modbus address, the UNICOS generator program has been modified to create output in the format required by EPICS. The mapping is then loaded into an IOC containing the modbus interface and EPICS sequencer programs that implement mass loading of parameters ("recipes").

The use of CERN's proprietary time stamp protocol (TSPP) is required when interfacing to UNICOS projects on Siemens PLCs. For Schneider PLCs, it is optional and improves the precision of the timestamp of register changes

beyond the resolution achieved with polling via modbus. This protocol has not been implemented for the EPICS port, yet.

UNICOS Widget Set

The main part of the port consists of the implementation of the UNICOS widget set for CSS. UNICOS widgets have a well-defined behavior that had to be ported. Each widget needs to access several PLC registers, i.e. several EPICS PVs to determine how to display itself. Due to the complexity of this, the implementation has been done in the form of an Eclipse module implementing custom widgets for BOY.

Implementing widgets listening to changes in more than one PV is easily possible. The main part of the implemented Java code then consists of the implementation of the display logic and the actual widget drawing. The UNICOS-defined dialogs (faceplates) used to interact with the objects in the system are implemented as static OPIs, making extensive use of OPI macros to implement the system-specific behavior. During the template OPI generation step, all widgets are created in a way that defines all required macros.

Hierarchy Displays

UNICOS control systems are modelled as a hierarchy of objects. E.g. analog alarm objects usually have an analog sensor as a child, and may themselves be the child triggering the interlock on a PCO. To browse this hierarchy from within CSS, static OPIs representing the siblings of each object are created from the output created by the UNICOS generator by a python script.

Alarm Handling

Since the PLC runs its own standalone control logic, all alarm logic is also located there. Generation, blocking, and persistence of alarms is handled independently from the supervision layer. The alarm state of each object is just signalled via bits in the status word, and acknowledgement of an alarm from the supervision layer is sent as a request to the PLC. To integrate the alarms into a wider EPICS/BEAST alarm system, the respective status bits are read and used to activate non-latching BEAST alarms that display in the CSS alarm screens. In parallel, according to the UNICOS standard, each UNICOS widget also display the alarm state.

The list of alarm objects to be considered is output by the UNICOS generator program.

Archiving

The UNICOS object definition contains useful information for the archiving process, such as the unit, or archiving deadband. The EPICS generator script puts these data in the corresponding fields of each PV. We use the EPICS Archiver Appliance [4] configured to honor the deadband put in the ADEL field.

RESULTS

The control system of the IBelle CO₂ cooling plant has been developed at CERN using UNICOS methodology with

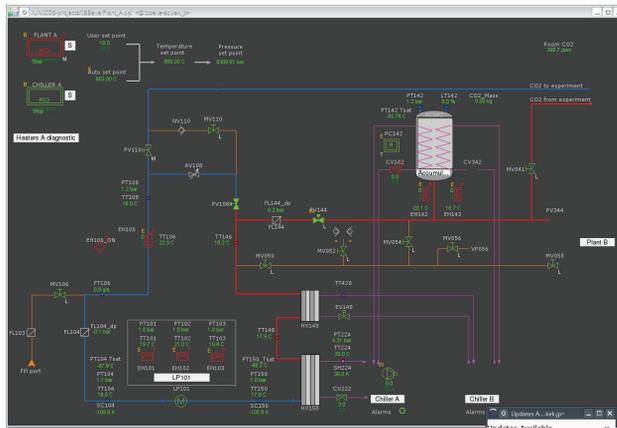


Figure 3: CSS displaying a UNICOS control screen.

the CPC6 framework, and WinCC OA. The plant is used to cool the two innermost subdetectors of the Belle II experiment, the pixel and strip vertex detector (PXD and SVD). Their control systems are based on EPICS, therefore the IBBelle control system had to be ported.

The methods presented in this paper have been used for the porting. After running the UNICOS generator program, 5259 PVs have been created for the 1051 UNICOS objects in the system.

The plant has been installed in its final position at KEK, Japan, and is operated successfully with the EPICS port. A screenshot of the ported control system is shown in Fig. 3.

CONCLUSION

We presented the successful port of the CERN UNICOS control framework to use EPICS as the supervision layer. The tools used are CSS for operator displays, BEAST for alarm integration, and the Archiver Appliance for archiving. For all tools, the required input files are produced by the modified UNICOS generator program and a set of Python scripts. A first CO₂ cooling plant controlled with UNICOS/EPICS is in production mode.

ACKNOWLEDGEMENTS

This work has been supported by the German Federal Ministry of Education and Research (BMBF).

REFERENCES

- [1] *Experimental Physics and Industrial Control System*. <http://www.aps.anl.gov/epics>
- [2] *Control System Studio*. <http://controlsystemstudio.org>
- [3] Mark Rivers. *Driver Support for Modbus Protocol under EPICS*. <http://cars9.uchicago.edu/software/epics/modbusDoc.html>
- [4] *The EPICS Archiver Appliance*. https://slacmshankar.github.io/epicsarchiver/_docs