UNICOS CPC6: AUTOMATED CODE GENERATION FOR PROCESS CONTROL APPLICATIONS[*]

B. Fernández Adiego, E. Blanco Viñuela, I. Prieto Barreiro, CERN, Geneva, Switzerland

Abstract

The Continuous Process Control package (*CPC*) is one of the components of the CERN Unified Industrial Control System framework (*UNICOS*) [1]. As a part of this framework, UNICOS-CPC provides a well defined library of device types, a methodology and a set of tools to design and implement industrial control applications. The new CPC version uses the software factory UNICOS Application Builder (*UAB*) [2] to develop CPC applications. The CPC component is composed of several platform oriented plug-ins (PLCs and SCADA) describing the structure and the format of the generated code. It uses a resource package where both, the library of device types and the generated file syntax, are defined. The UAB core is the generic part of this software, it discovers and calls dynamically the different plug-ins and provides the required common services. In this paper the UNICOS CPC6 package is introduced. It is composed of several plug-ins: the Instance generator and the Logic generator for both, Siemens and Schneider PLCs, the SCADA generator (based on PVSS) and the CPC wizard as a dedicated plug-in created to provide the user a friendly GUI. A tool called UAB Bootstrap will manage the different UAB components, like CPC, and its dependencies with the resource packages. This tool guides the control system developer during the installation, update and execution of the UAB components.

INTRODUCTION

In this paper, we introduce the UAB CPC6 (*UNICOS Application Builder - Continuous Process Control*) component as a part of the UNICOS (*Unified Industrial Control System*) framework. UNICOS is a control system framework, developed at CERN (*European Organization for Nuclear Research*), designed to implement control system applications. This framework provides a methodology, an object library and a set of tools to generate the control code for these applications. A UNICOS component or package is a part of the UNICOS framework which uses the UNICOS methodology to create a specific type of applications, provides its own object library and uses the UNICOS generation tools to obtain the control code for these applications. Currently there are several UNICOS components, used for different accelerator systems, such as CPC, CIET (*Cryogenics Instrumentation Expert Tool*), QPS (*Quench Protection System*) and SURVEY (Control system for the magnets alignment) [3] (See Fig. 1).

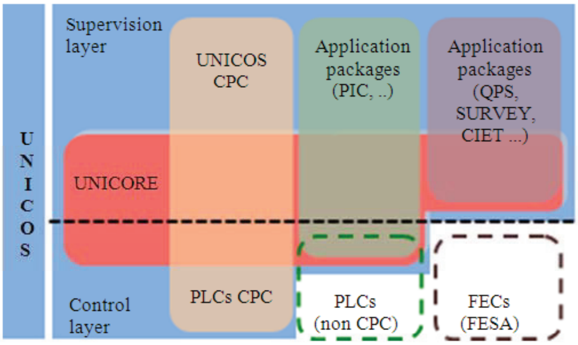


Figure 1: UNICOS Overview.

The CPC component has been designed to develop industrial control system applications for continuous processes. It has been used for more than ten years in continuous processes like the LHC (*Large Hadron Collider*) cryogenics system, the gas systems of the LHC experiments, interlock system and lately cooling & ventilation and vacuum systems [4]. The CPC package creates applications focused in the two upper layers of a control system (Supervision and Control) and sets the communication between them automatically. To achieve this, the CPC objects Baselines have been developed at the SCADA (Supervisory Control And Data Acquisition) and the control levels. The control layer is PLC (Programmable Logic Controller) based, currently it has been developed for the Siemens S7 and Schneider platforms and for CoDeSys (Controller Development System) which makes it platform independent, while the SCADA layer has been developed for PVSS, nowadays called WinCC OA (WinCC Open Architecture).

UNICOS CPC6

The UNICOS-CPC package presents three significant modifications from the previous version: the methodology, the objects and a new generation tool called UAB.

The UNICOS methodology has been improved providing a set of documents and some well-defined steps to develop CPC control applications. Following this methodology, the CPC user will be able to transform the process knowledge into a model based in a hierarchy of CPC objects (See Fig. 2), and implement the control system using that information. These objects have been improved

[*] Work partially funded by the Spanish Ministry of Science and Innovation

from the previous version, including new functionality and more flexibility for the users and operators [5].

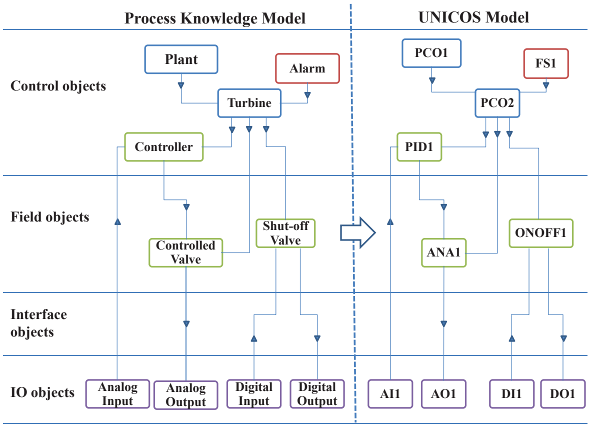


Figure 2: Example of CPC objects hierarchy.

This methodology contains the following steps: (See Fig. 3).

- Functional analysis: the process engineer transfers the process knowledge to the UNICOS Functional Analysis document.
- UNICOS Logic design: the process engineer and the control engineer work together to define the behavior of the process in the UNICOS language.
- Specification file filling: all the UNICOS objects are defined and parametrized using an Excel/XML file.
- Automatic generation: the instance and logic code for the PLC and the configuration file for WinCC OA are generated using the automatic generation tools.
- Logic completion: complete the automatic generation of the logic if necessary, following the UNICOS Logic design.
- Control code compilation.
- Application tests.
- Commissioning.
- Operation.

USING AUTOMATIC GENERATION TOOLS

The use of automatic generation tools allows development of control system applications with a high level of abstraction, decreasing the development, configuration and commissioning time and producing well-structured control code. For this new version, one of the most important requirements was to include a more flexible, extensible, consistent and user-friendly automatic generation tool. As result of these requirements the UAB has been developed. This tool generates the control code extracting information from the UNICOS objects definition and the process knowledge included in the specification file [2].

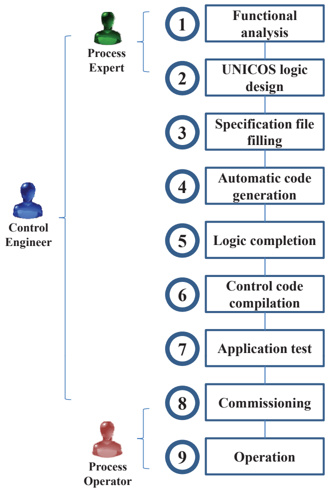


Figure 3: UNICOS-CPC6 methodology.

Architecture

The UAB architecture is composed of three main parts: the UAB core, the UAB plug-ins and the resource package (see Fig. 4). These three parts have different functionalities and life cycles.

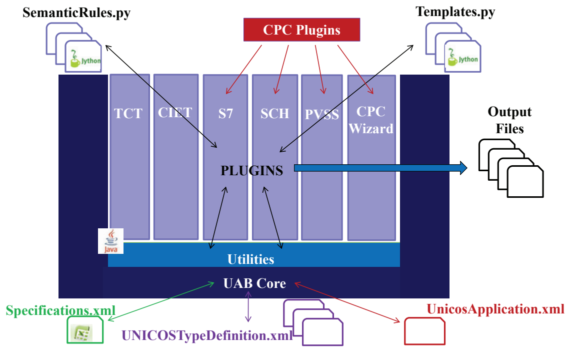


Figure 4: UAB Architecture.

The UAB has been developed in the Java language as well as other technologies like the Jython (Pyhton for Java) scripting language to define the syntax and the contents of the output files. The format for the configuration files is XML (*eXtended Markup Language*) and the technologies used to process these files are JAXB (*Java XML Binding*) and JXPATH (*XML Path Language for Java*). Apache Maven is the selected technology to build and manage the tool.

All the UNICOS objects, as the CPC objects, are defined as XML files (*DeviceTypeDefinitions*), where the object structure is defined (Inputs, Outputs, Frond-End Parameters, SCADA Parameters, etc.). For this purpose, the so-called *UNICOS Metamodel* has been designed. This Metamodel is an XSD (*XML Schema Definition*) file used to describe and limit the contents of the DeviceTypeDefinitions [6].

UAB Core

The UAB core is the main and generic part of the tool. It is platform independent and provides the common services required by most of the plug-ins (for example User report, logging, etc.). It also has other functionalities, for example dynamically discovering the different plug-ins and asserting their validity, loading the UNICOS Project data information, connecting the plug-ins with the external files, etc.

UAB Plug-ins

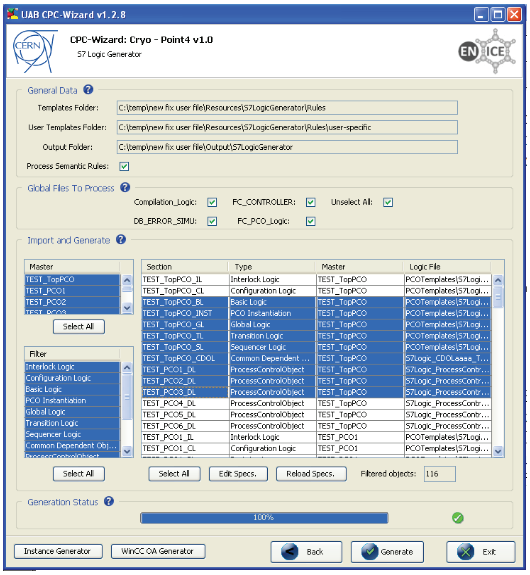
A plug-in is the part of UAB where the structure and the format of the generated files are defined. It is platform dependent (i.e. Siemens, Schneider, etc.) and its main function is to generate these files.

Currently, the CPC package contains the following plug-ins:

- Siemens Instance Code Generator: it generates the instance and communication files in SCL (*Structured Control Language*) format and the symbols files containing the address mapping.
- Siemens Logic Code Generator: it generates the control logic files in SCL format.
- Schneider Instance Code Generator: it generates the instance and communication files in XML format.
- Schneider Logic Code Generator: it generates the logic files in XML format.
- WinCC OA Code Generator: it generates the configuration file for WinCC OA in TXT format
- CPC Wizard: it is a dedicated plug-in that provides a friendly GUI to drive the generation of CPC applications. It prompts the user for the mandatory data of the application, validates the user data and triggers the execution of the selected plug-ins with the specified parameters. The wizard's panels can be customized through an XML file and a set of predefined components (like text editors, radio buttons, etc.) (See Fig. 5).

These plug-ins use the services provided by the UAB core to perform the following tasks:

- Read the data from the input sources, like the UNICOS specification file or the DeviceTypeDefinitions.
- Process the semantic check rules to validate the input data.
- Execute the Jython scripts.
- Generate the output files.



CPC-Wizard: Cryo - Point4 v1.0
S7 Logic Generator

Templates Folder: C:\temp\new fix user file\Resources\S7LogicGenerator\Rules

User Templates Folder: C:\temp\new fix user file\Resources\S7LogicGenerator\Rules\user-specific

Output Folder: C:\temp\new fix user file\Output\S7LogicGenerator

Process Semantic Rules: ☑

Compilation_Logic: ☑ FC_CONTROLLER: ☑ Unselect All: ☑

DB_ERROR_SIMU: ☑ FC_PCO_Logic: ☑

Master
TEST_TopPCO
TEST_PCO1
TEST_PCO2
TEST_PCO3

Filter
Interlock Logic
Configuration Logic
Basic Logic
PCO Instantiation
Global Logic
Transition Logic
Sequencer Logic
Common Dependent Obj...
ProcessControlObject

Section	Type	Master	Logic File
TEST_TopPCO_IL	Interlock Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_CL	Configuration Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_BL	Basic Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_INST	PCO Instantiation	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_GL	Global Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_TL	Transition Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_SL	Sequencer Logic	TEST_TopPCO	PCOTemplates\S7Logi...
TEST_TopPCO_CDOL	Common Dependent ...	TEST_TopPCO	S7Logic_CDOLaaaa_T...
TEST_PCO1_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO2_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO3_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO4_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO5_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO6_DL	ProcessControlObject	TEST_TopPCO	S7Logic_ProcessContr...
TEST_PCO1_IL	Interlock Logic	TEST_PCO1	PCOTemplates\S7Logi...
TEST_PCO1_CL	Configuration Logic	TEST_PCO1	PCOTemplates\S7Logi...

Filtered objects: 116

Figure 5: Wizard panel for the Siemens Logic generator.

The plug-ins are completely independent and it is possible to launch them through the wizard or one by one using the XML editor called FESA (*Front-End Software Architecture*) General editor from the FESA framework. Apart form the CPC plug-ins, more plug-ins have been developed in UAB. For example, for the CIET component the instance and WinCC OA configuration plug-ins have been created.

UAB Resources Package

The UAB resources package is the part of the tool where the CPC objects and the contents of the output files are defined, it also contains the semantic check rules used to validate the user inputs during the definition of the project. The CPC resources package contains the following list of elements:

- Generation Templates: these templates, written in Jython scripting language, define the syntax and the contents of the generated control code. There are five types of generation templates: the Siemens instance templates, the Siemens logic templates, the Schneider instance templates, the Schneider logic templates and the WinCC OA templates.
- DeviceTypeDefinitions: the XML files which contain the definition of the CPC objects.
- Baselines: the baseline is the definition of the CPC objects in PLC code and the additional common code necessary to make the application run. It uses the structure defined in the DeviceTypeDefinition and contains the behavior of the objects. Currently the CPC package provides the Siemens and Schneider Baselines and also the UnCPC package which represents the WinCC OA behavior of the CPC objects.
- Semantic Check rules: these are special kinds of

templates used to check the specification file.

- Input/Output Commissioning template: it's a special kind of template designed to generate a IOCommissioning file to help the process and control engineer during the commissioning of the system.

UNICOS MANAGEMENT

The special architecture of UAB, where the software is split in several packages with different life cycles requires an appropriate mechanism to manage the different software versions (for example, while the UAB Core will be very stable, the resource package is susceptible to be modified frequently, changes such as adding new devices, modifying existing devices or modifying the syntax of the output files). For this purpose, the UAB Bootstrap has been created. This tool, developed in Java, is used to manage the different UAB components and its resources packages. The main functionalities of this tool are:

- Upon first installation of UAB, it offers to download the last version of the available UAB components.
- Check for updates of the installed components and its compatible resource packages.
- Download/install new UAB components.
- Unique entry point to execute the UAB components installed.

The different pieces of software that compose UAB (core, plug-ins, resources) are packaged in different artifacts and deployed in a repository manager (Nexus). When the Bootstrap is executed, it queries the repository manager to discover new components, new versions of the installed components and new versions of the resource packages. When a new version is available on the server, it will notify the user and offer to download the new software.

To achieve this, the Bootstrap uses the *Nexus REST API* and the *Aether library*. The repository manager provides the REST API that can be used externally to query Nexus for available artifacts or test if any of the available artifacts has a version number higher than the installed ones. Aether is a general purpose library for interacting with artifact repositories. It provides functionalities to specify the repository locations, dependency resolution between artifacts and artifacts download.

Figure 6 shows the different steps to create an application for a CPC user. The UAB Bootstrap packs the selected CPC component version and the compatible CPC resources package from the Nexus server and installs the UAB software in a local machine. Thus the user will create a CPC application using the CPC wizard installed.

CONCLUSION

The UAB CPC6 component has been developed to add several improvements for developers and users (control system engineers).

The developer, using UAB, will obtain more flexibility and performance, allowing to add new CPC objects easily, new plug-ins for new platforms (for example the CIET plug-ins), templates or semantic check rules.

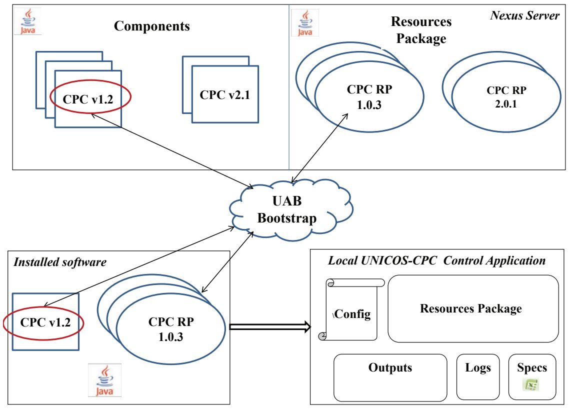


Figure 6: UAB Management.

For the user, this new version offers a user-friendly graphic interface to develop CPC applications and a set of services like a powerful version management which guaranties the maintenance of these applications, logging, semantic check verifications, etc. The user can also add new objects, templates or check rules to the resources package.

REFERENCES

[1] Ph. Gayet, R. Barillère. "UNICOS a framework to build industry like control systems: Principles & Methodology".*10th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systemns.* Genève (Switzerland), 10-14 Oct 2005.

[2] M. Dutour. "Software Factory Techniques applied to process control at CERN". *11th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systemns.* Tennessee (USA), 15-19 Oct 2007.

[3] Hervé Milcent, Enrique Blanco, Frédric Bernard, Philippe Gayet. "UNICOS: AN OPEN FRAMEWORK". *12th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systemns.* Grenoble (France), 12-16 Oct 2009.

[4] D. Willeman, E. Blanco, B. Bradu, J. Ortola. "UNICOS CPC new domains of application: Vacuum and Cooling & Ventilation". *13th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systemns.* Grenoble (France), 10-14 Oct 2011.

[5] E. Blanco, A. Merezhin, B. Bradu, B. Fernández Adiego, D. Willeman, J. Rochez, J.M. Beckers, J. Ortola Vidal, Ph. Durand, S. Izquierdo Rosas. "UNICOS Evolution: CPC Version 6". *13th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systemns.* Grenoble (France), 10-14 Oct 2011.

[6] B. Copy, R. Barillere, R.N. Fernandes, B. Fernandez Adiego, I. Prieto Barreiro. "Model Oriented Application Generation for Industrial Control Systems". *13th ICALEPCS Int. Conf. on Accelerator and Large Expt. Physics Control Systems.* Grenoble (France), 10-14 Oct 2011.