UNICOS A FRAMEWORK TO BUILD INDUSTRY LIKE CONTROL SYSTEMS: PRINCIPLES & METHODOLOGY

Ph. Gayet, R. Barillère,
CERN, Geneva, Switzerland

ABSTRACT

UNICOS is a CERN framework developed to produce control applications for three-layer industrial control systems (Fig. 1). UNICOS provides developers with means to develop full control applications and operators with ways to interact with all items of the process from the most simple (e.g. I/O channels) to the high level compounded objects (e.g. a sub part of the plant). In addition UNICOS offers tools to diagnose the process and the control system.

UNICOS proposes a method to design and develop the control applications. This method is based on the modelling of the process in a hierarchy of objects (I/Os, actual devices and more abstract control objects). These objects are used as a common language by process engineers and programmers to define the functional analysis of the process.

In addition to the method, tools have been produced to automate the instantiation of the objects in the supervision and process control layers and generate skeletons of the Programmable Logic Controller (PLC) programs. The control code production associated to this methodology is generally data-driven and for some projects a model driven software production has been implemented.

INTRODUCTION

Use of industrial control COTS for CERN applications

Nowadays many industrial control system components are used in the high energy physics community. Several CERN accelerators and experiment systems are controlled with industrial Component Off The Shelves (COTS) only (e.g. LHC cryogenic system, the LHC vacuum system, LHC experiments gas system, LHC experiments detector safety, and LHC experiments magnet).

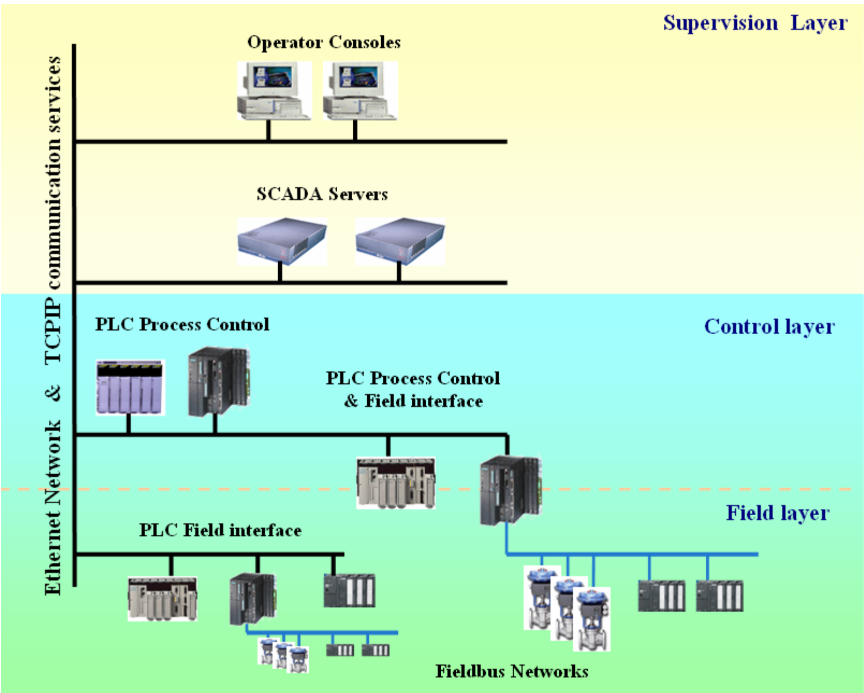

Figure 1: Industrial three-layer control system

Theses components are distributed across three layers: The field layer contains the process sensors and actuators connected to the control system via PLC I/O boards and/or field buses. The control logic is implemented in PLCs in the control layer. The supervision layer provides operators with monitoring and command facilities by means of Supervision Control And Data Acquisition (SCADA) systems.

Integration of the three layers

One of the major problems faced by such type of control architecture is the heterogeneity of the different layers, the components being provided by different suppliers. Despite communication standards, there is almost no product on the market tightly integrating SCADA, PLC and Field buses. Although industrial Distributed Control Systems (DCS) combine the three layers, their use induces much higher costs and implies proprietary solutions incompatible with the integration into the accelerator/detector control systems complex

Thanks to the existing standards, software companies have incorporated in their offers solution to integrate the three layers. But each integrator has its own solution. Hence when laboratories outsource the development of their control system, they either become dependent of external companies for the maintenance of strategic systems or impose to the laboratory control teams to develop competences in many incompatible solutions.

The purpose of UNICOS is to offer a control model integrating COTS of the three layers. UNICOS can be applied to hardware and software items of many suppliers. The model is implemented by libraries components and tools for the CERN-wide recommended hardware and software. Then, provided they received a proper training, the control system can be produced by laboratory or detectors collaboration teams. After an appropriate call for tender, it can also be developed by an external company. The integration into the accelerator/detector whole control system is guaranteed as well as the internal resource knowledge for maintenance.

In addition, to ease the software production, UNICOS proposes a process analysis methodology. This methodology aims at allowing the production of a functional analysis by the process engineers which can be directly translated into PLC codes by the programming team. Moreover during this analysis the operator interactions with the process are studied and included in the developed control software application.

PROCESS MODELLING

The IEC61512-1 NORM

A common approach to process modelling is to break down the process in a hierarchy of modules. This principle in applied to the physical model of the IEC61512-1 (Former ISA 88.00) Norm for batch control (fig. 2). One part of the model covers the production from the management of the entire enterprise to the adaptation of a unit into several batch productions. The other part (Unit, Equipment modules, Control modules) is applicable to continuous process control:

- The Unit is made of Equipment and Control modules. The modules that make up the unit may be configured as part of the unit or may be acquired temporarily to carry out specific tasks. One or more major processing activities can be conducted in a unit. It combines all necessary physical processing and control equipment required to perform those activities as an independent equipment grouping. It is usually centred on a major piece of processing equipment.

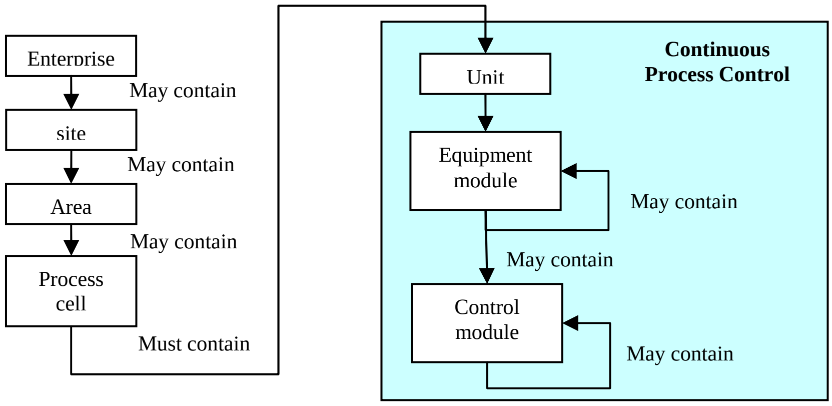

Figure 2: IEC61512-1 Physical Model & Continuous Process control Modelling

- The equipment module can carry out a finite number of specific minor processing activities. It combines all necessary physical processing and control equipment required to perform those activities. It is usually centred on a piece of processing equipment. Functionally, the scope of the equipment module is defined by the finite tasks it is designed to carry out.
- A control module is typically a collection of sensors, actuators, other control modules, and associated processing equipment that, from the point of view of control, is operated as a single entity. A control module can also be made up of other control modules. For example, a header control module could be defined as a combination of several on/off automatic block valve control modules.

THE UNICOS OBJECT TYPES

Hence as for IEC61512-1, UNICOS proposes to break down the process in a hierarchy of objects were each object as a unique parent. These objects can be sorted in three categories (fig.3):

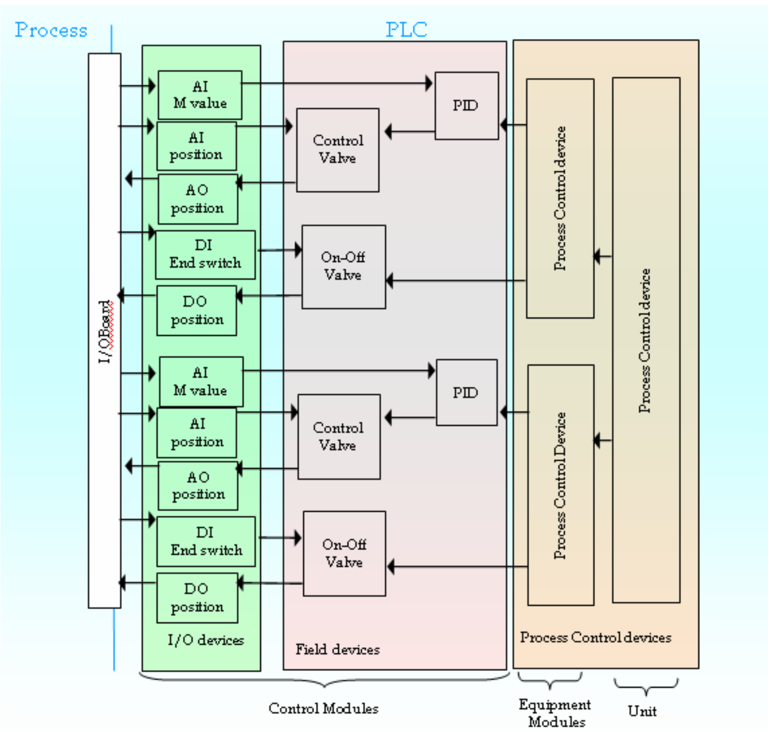

Figure 3: The UNICOS Object Hierarchy

- The I/O objects provide the interface to the plant. They link the sensors and actuators to the control system. Here the digitalisation process is performed and some basic treatments are carried out. Input/Output (I/O) channels shall be accessed through such objects exclusively.
- The field objects, they are the images of the hardware elements such as valves, heaters, motors, and others or they perform control task like such as PID loop. Process values are acquired through Input objects and process outputs are set via Output objects.
- The process control objects (PCO), responsible for the control of equipment units grouping several Field objects and/or other process control objects coping with subparts of the specific equipment.

The I/O and field objects are obviously Control Modules, whereas the PCO can be considered either as equipment modules or as unit according to the level of complexity they handle.

Supervision & Control integration

The main issue in the integration is the relation between the supervision and the control layers. The link between the control and the field layer is indeed usually implemented by PLC's firmware (at least in those selected at CERN).

The UNICOS objects are implemented in the PLCs and each of them is provided with a proxy in the supervision layer. The UNICOS middleware guarantees the synchronisation between the PLC objects and their respective proxies. In particular, the result of each command or process changes is transmitted from a PLC object to its proxy to be displayed and archived. Operators can then check that the PLC has treated their requests.

All UNICOS objects classes have been designed according to the same principles (fig. 3).

The PLC objects implement the process behaviour. The object internal logic receives:

- Information from the process (process inputs: analogue or binary values from sensors and statuses of other objects);
- Commands from operators;
- Commands from the control logic (Auto Requests from parent PCO);
- Configuration parameters set during the programming phase and accessible for modification by a program specialist.

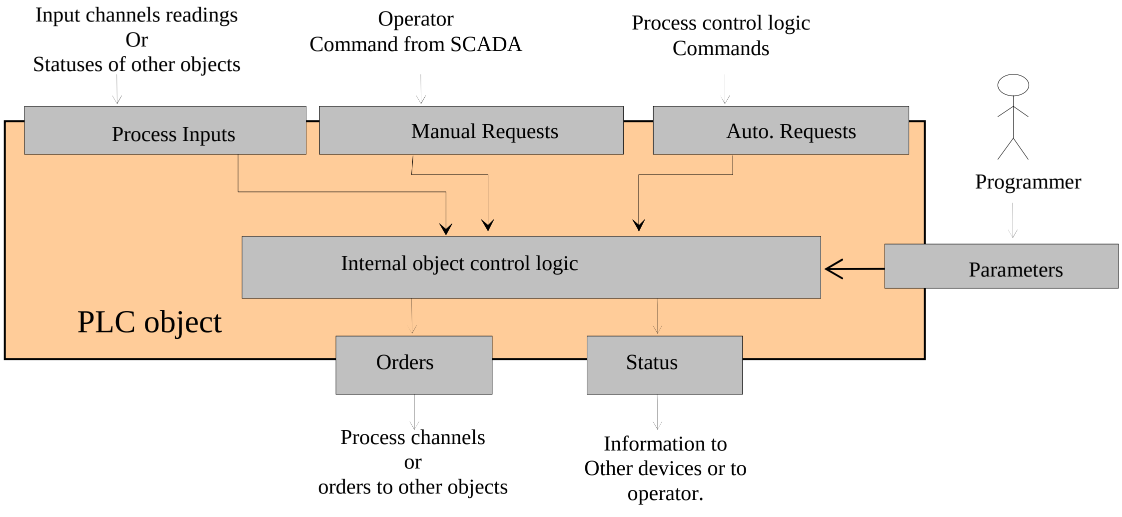

Figure 3: PLC object model

According to the internal states of the object (“auto”/ “manual”, interlocked, started or stopped etc.), the PLC object control logic processes the inputs & requests and sends orders either to the process outputs or child objects. The PLC object logic publishes the status of the object to inform the other objects or operators via the communication middleware.

The SCADA object proxies provide operators with feedback and interaction facilities (fig. 4) by means of icons (widgets) and dedicated panels (faceplates) developed for each object class.

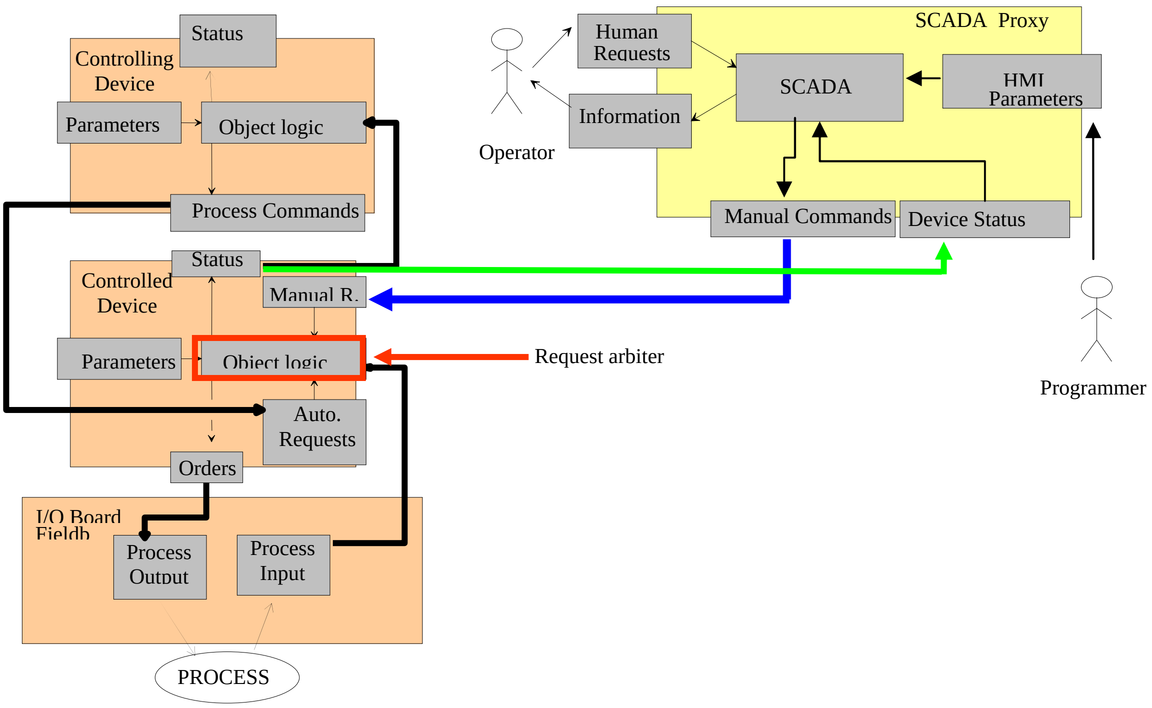

Figure 4: HMI PLC integration

UNICOS PCO: A single model for the Unit & Equipment Module

Whereas the I/O and field objects are fully generic and can be re-used as such by any application, the PCOs which are the nodes of the object hierarchy must be versatile enough to cope with complex and application specific process logic. They provide indeed developers with placeholders for the application specific logic:

- Conditions for the Controlled Stop completion (used to stop the object in a controlled way).
- Configurable interlock conditions to stop PCOs in case of severe process anomalies: Start interlock (avoid start-up but inducing no stop), Temporary-stop (to stop in case of anomalies but allow restart when it disappears) and Full-stop (to stop in case of anomalies but allow restart only after an operator acknowledgement).
- Configurable feedbacks: The On/Off status of a PCO is the result of the combination of status of its child objects and process inputs.
- Global Logic: to implement global parameters computation or a finite state machine.
- Dependant object logic: Piece of code which drive each child object of a given PCO. It define the “auto” requests to be sent according to process inputs, the status of the PCO and if any, the finite state machine.

In addition to the internal object logic described above, the PCO one’s implements:

- Propagation of the alarm acknowledgements and transitions to auto mode toward their children;

- Evaluation of the following PCO specific properties which will be used in the global and dependent objects logic: Interlocks dependant start; Controlled stop request; Option Modes handling (to set the PCO in different "options mode" such as cool down, warm-up etc.)

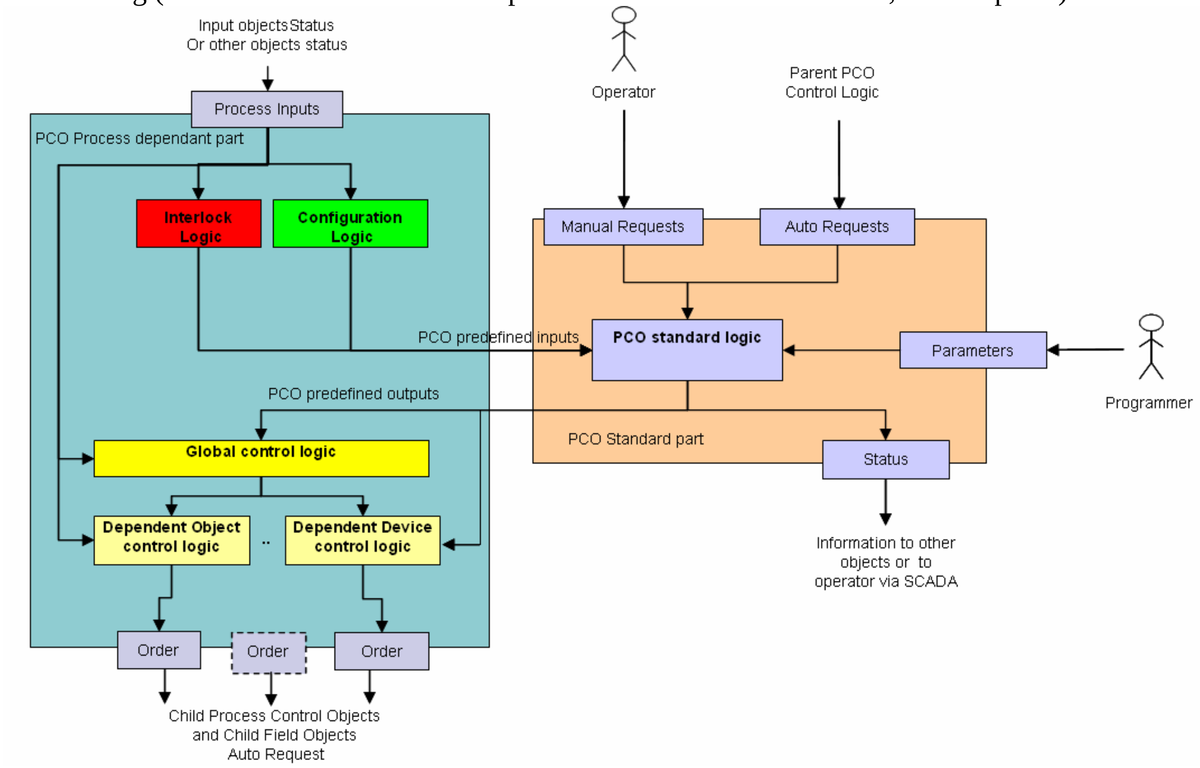

Figure 5: PCO Object model

APPLICATION PRODUCTION & MAINTENANCE

A set of components and data-driven tools supporting the UNICOS model have been produced. They help the application developers in all the phases of their software production.

Extension of UNICOS object type library

A first set of field objects have been developed to cover the specific needs of the LHC Cryogenic system [1]. Prior to the use of UNICOS, the process engineers have to verify the completeness of the UNICOS process field object library. In case of missing types, new object types have to be developed in SCADA and PLC layers and the generation tools adapted accordingly. For instance, this list has been extended for the LHC GCS project [4].

Functional analysis and PCO model

With UNICOS, process engineers write process functional analysis by defining a hierarchy of PCO and using the PCO model. Hence, as in *IEC61512-1* they have first to identify the different Units and Equipment modules (PCO), grouping together control modules (field objects). This is the most critical phase as there are no general solutions. For fluids systems, a good approach is to identify independent circuits performing different and independent tasks. Then, once the break down and the hierarchy established, for each PCO, Process engineers have to define: the interlocks and feedback conditions, the option modes if they are requested, the finite state machine or global parameters computation and for each child objects the specific dependant logic.

UNICOS users are provided with a template form [1] to capture the Functional Analysis. Once filled, it becomes the programming specification of all routines to be linked to the PCO logic placeholders.

By going through this phase, process engineers are in fact covering a task which was traditionally devoted to software engineers. They are identifying the means by which operators will interact with the process.

Objects specification

In addition to the Functional analysis, the process team has to define the complete list of parameters of each I/O and field objects. This is achieved by filling the provided EXCEL object type spreadsheets. These parameters include process information for the PLC and configuration data for the supervision proxies.

Instantiation generation

The first phase of the coding can then start: the PLC objects and their PVSS proxies have to be instantiated in the two layers. The Instance Generator, a data-driven tool, produces configuration files from the object list spreadsheets. These configuration files are imported into the PLC and SCADA, the middleware is automatically configured. From this stage, it is already possible to start the commissioning of the installation from the supervision down to the sensors and actuators.

Logic and specific HMI production/generation.

The second coding phase consists in producing the application specific PLC code. The logic generator tool is used to provide developers with PLC program skeletons which are compliant with the PCO hierarchy. From the functional analysis the programmer can directly start the coding of the logic into the PLC routines following precise guidelines and using the PLC skeletons as placeholders of his specific code.

No code development is required for the production of the supervision layers. Developers produce their synoptic views with a drag and drop drawing tool.

If the process modelling reveals the possibility to have replicated equipment modules or when the same type of equipment has to be deployed several times, it is possible to replace the manual production phase by an automated code production.

When equipment modules are replicated, the interlock logic, configuration logic and global dependent child objects logic can be written in “PLC like” languages files. These so called “include files” are used by the logic generator which will produce complete PLC code instead of generic skeletons.

One can go even further: When similar but not fully identical equipment as to be deployed, as it is the case for the LHC experiments gas systems’, a model driven approach can be used. A model-based generator can fill the object spreadsheets used by the instance and logic generator. This possibility coupled to the use of include files allows a complete model-driven code production.

Finally the interactive production of supervision layers (views, trending, etc…) can also be replaced by an automatic configuration from a model-based generator [3].

These advanced production tools are used for the LHC experiment gas control systems and will be adopted for the LHC magnet sector cryogenic.

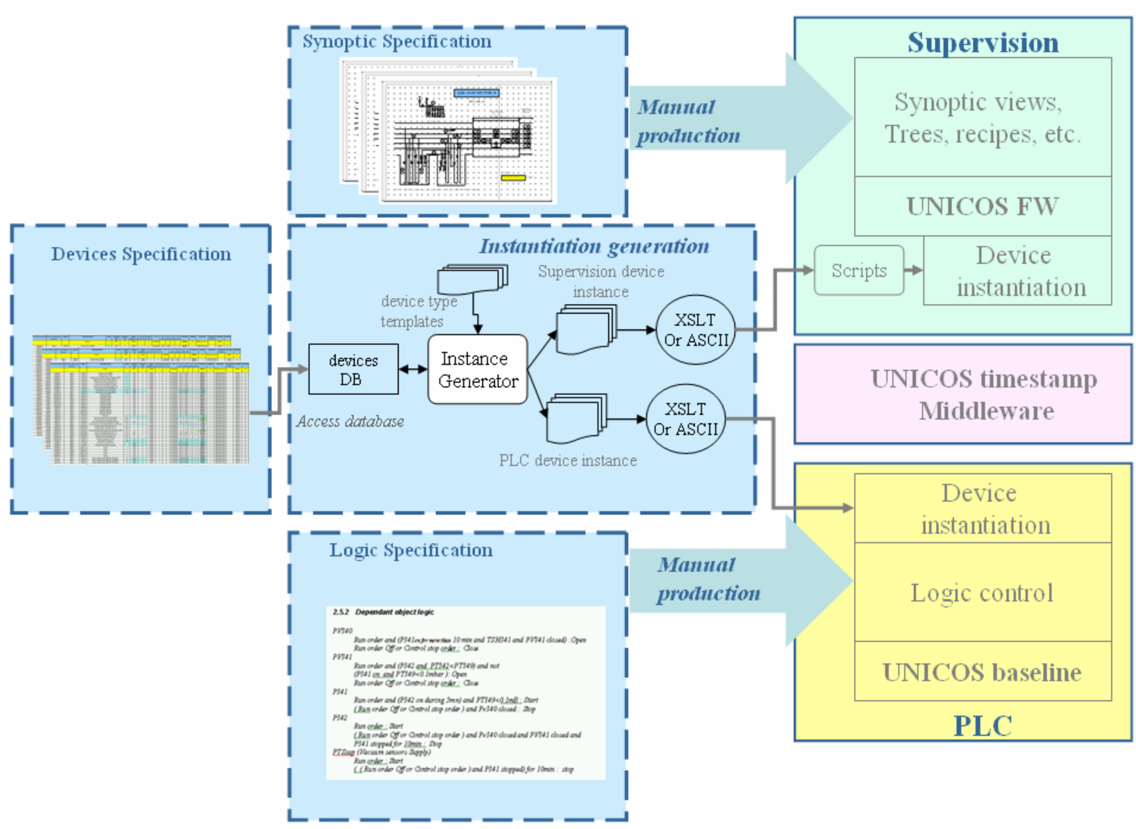

Figure 6 : UNICOS Basic Software Production

Maintenance.

The Functional analysis and strong coding guidelines are producing a homogeneous code structure and contents. This allows the maintenance of a large volume of code by a small team. In case of functional analysis correction during the commissioning it is possible and easy for this team to track and maintain the code.

AVAILABILITY OF UNICOS FRAMEWORK

The UNICOS framework has been deployed at CERN on Schneider and Siemens PLCs and on the PVSS SCADA system. The UNICOS PLC framework includes libraries of the object routines (Schneider DFB and Siemens FB) and application templates for each type of PLCs including the communication middleware. The UNICOS PVSS framework contains user interface components libraries (widgets & faceplates for each object type) and a complete user interface environment described in [5].

In addition to these libraries, tools have been produced to automatically generate PLC objects and their PVSS proxies from a single database. The tools also provide skeletons of PLC codes for both types of PLCs.

CONCLUSION

Since its first application commissioned in July 2001, UNICOS principles have prove to be independent of their platform. They have been ported in two successive SCADA and in several PLC programming environments.

UNICOS has been used to produce several CERN control applications. The code production has been done either by CERN teams (LHC GCS, Experiments magnet and Cryogenic), LHC detectors collaboration (ATLAS Argon cryogenic) or partly outsourced to an external company (Accelerator cryogenic).

UNICOS has some limitations, compared to usual PLC programming; it implies extra memory usage and additional CPU load. Hence this method is not suitable for time critical process (<10ms).

UNICOS-based applications proved to be very efficient tools for operators. Operators can interact with the any application objects through a standardized interface. This possibility offers flexible troubleshooting facilities during tests or in case of equipment failure. It also increases availability of the control system in case of unforeseen hardware failure.

Moreover the easy and efficient creation and modification of process views does not involve programming knowledge, it can then easily be transferred to the operation crew.

The UNICOS concepts are complex and their use and adoption by the process engineers requires an appropriate training. But once mastered the specification templates become a powerful tool to produce a functional analysis and are easy to be translated into PLC code by programmers. To ease this training the UNICOS team has to produce more complete and more didactic documentation.

Very rigorous programming guidelines have been established, they reduce the programmer creativity and freedom but they homogenize the software production and ease the maintenance by any team member (e.g. the outsourced application are now maintain by CERN team).

The basic UNICOS data driven tools alleviate developers from painful and error prone tasks such as object generation in PLC and SCADA and the communication configuration. The advanced model-based production tools are mature and promising.

Acknowledgement:
The authors would like to thanks all members of the UNICOS and LHC GCS teams for the present achievements. And the two teams are grateful to the UNICOS customers for their patience and understanding.

REFERENCES

[1] The AB UNICOS project, URL: http://ab-project-unicos.web.cern.ch/ab-project-unicos/
[2]G. Thomas et al., “LHC GCS: A framework for the production of 23 homogeneous control systems”, ICALEPCS’2005, Geneva, Switzerland, October 2005. [PO1.042-6]
[3]G. Thomas et al., “LHC GCS: A model-driven approach for automatic PLC and SCADA code generation”, ICALEPCS’2005, Geneva, Switzerland, October 2005. [FR2.3-6O]
[4] The LHC GCS project, URL: http://itcofe.web.cern.ch/itcofe/Projects/LHC-GCS/
[5] C.H. Sicard et al, Deploying the UNICOS industrial controls framework in multiple projects and architectures, ICALEPCS’2005, Geneva, Switzerland, October 2005[WE3.A2-6O]