**International Carpathian Control Conference ICCC' 2008**
Sinaia, ROMANIA
May 25-28, 2008

# AN AUTOMATIC APPROACH TO PLC PROGRAMMING FOR A LARGE SCALE SLOW CONTROL SYSTEM (BASED ON LHC CRYOGENIC DISTRIBUTION CONTROL SYSTEM)

**Czesław FLUDER[1], Juan CASAS-CUBILLOS[2], Marek CIECHANOWSKI[3], Paolo GOMES[2], Paweł JODŁOWSKI[3], Michał KLISCH[3], Mateusz SOSIN[3], Antonio TOVAR[2], Tomasz WOLAK[3], Łukasz ZWALIŃSKI[3]**
[1] Department of Electrical Drive and Industrial Equipment
AGH University of Science and Technology,
Kraków, Poland, fluder@kaniup.agh.edu.pl
[2] Accelerator Technology Department
CERN
Geneva, Switzerland
[3] Faculty of Physics and Applied Computer Science
AGH University of Science and Technology
Kraków, Poland

**Abstract:** The large scale of the slow control system poses a real challenge in Programmable Logic Controllers (PLC) code production. The Cryogenic Distribution Control System for the Large Hadron Collider (LHC - CERN, Switzerland) consists of around 15 000 repetitive sensors and actuators operated by more than 3800 Closed Control Loops (CCL). UNified Industrial COntrol System (UNICOS) and code generator developed at CERN have been a starting point for the automatic code production. The generation tools have been adapted and extended to the Cryogenic Distribution Control System requirements. Package of software validation tools has been built for input data verification and consistency check. Process control logic specification and tables with relationships between system components have been used to create several types of code templates. To cover all non-standard code parts new generation tools have been developed. The final software validation has been done using the developer machine before deployment.

**Key words:** control systems, code production, PLC programming, cryogenics, LHC

## 1 LHC Cryogenic Distribution Control System

Large Hadron Collider (LHC), the world's highest energy particle collider (Fig. 1), is now being completed in European Organization for Nuclear Research (CERN). This proton-proton collider housed in the 27 km circumference tunnel is designed to provide proton collisions at 14 TeV in the centre of mass. LHC will provide information about the structure of matter on terascale (energy range 1 TeV involved) and energy in the universe. To control the two circulating proton beams the LHC uses hundreds of superconducting magnets, electrical Distribution Feed Boxes (DFB) with superconducting leads that supply magnets and Accelerator Superconducting Cavities (ACS) to speed up particles. All components mentioned above have to be cooled down to temperature close to 0 K (-273$^{o}$C) with liquid helium redistributed around LHC ring through

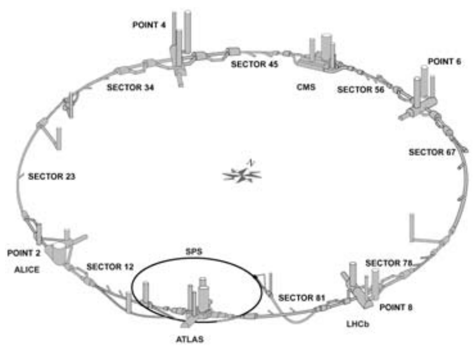


Fig. 1. Large Hadron Collider

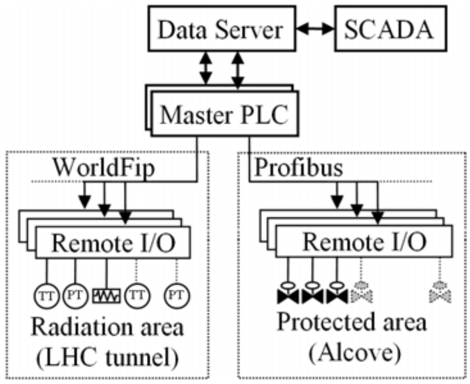


Fig. 2. One sector CDCS structure

cryogenic distribution line (QRL). Liquid helium supply is provided by gas storage, cold compressors and refrigerators located at points 1, 2, 4, 6, 8 and connected to QRL through interconnection boxes.

The Cryogenic Distribution Control System (CDCS), as a part of LHC cryogenics, is one of several systems required by LHC accelerator to be operational. CDCS consists of 18 most powerful SIEMENS® S7-400™ PLCs, two for each of 8 LHC sectors, and two for 16 ACSs. These PLCs control cryogenic instrumentation for up to 1800 superconducting magnets, 52 DFBs and 16 ACSs installed in the LHC tunnel. Approximately 15000 sensors and actuators have been installed and connected to the CDCS through Profibus and WorldFIP remote I/O modules (Fig. 2.).

Table 1. Sensors and actuators in CDCS

| Object type | Quantity |
|---|---|
| PT100 temperature sensor | 4353 |
| CX temperature sensor | 2658 |
| Baumer® pressure sensor | 854 |
| Superconducting helium level gauge | 436 |
| On Off valves | 680 |
| Analog valves (0-100%) | 2606 |
| Analog heaters (0-100%) | 2361 |
| Other On Off and Analog equipment | ~1000 |

Each CDCS instrumentation element has common characteristics (associated with its function and type) but specific parameters correlated with its instance. All these data are stored and described in several relational databases e.g.: the Layout database (containing description and location data for all objects), the Manufacturing and Test Equipment database and the Thermbase database (with individual calibration data for all sensors).

CDCS comprises all sensors and actuators in connection with PLC programs, containing up to 3800 Closed Control Loops (with modular PID algorithm) and 12-phase sequential control with process logic. CDCS is classified as real-time slow control system where required sampling period is longer than 100 ms [Brüning et al., 2004]. In principle the single PLC program contains around 250 000 lines of code written in Structured Control Language (SCL) that constitute up to 3.2 Mbytes of machine code and its cycle takes about 500 ms.

## 2 CDCS software production methodology

### 2.1 UNified Industrial COntrol System (UNICOS)

UNICOS, the starting point for software production, provides components, methodology and tools to design, build and program industrial based control systems. UNICOS environment supports all three control system layers: supervision (SCADA: PVSS), control (PLC: Siemens S7 and Schneider) and field layer. The package of programming tools includes: baseline library (with modular PID algorithm), code generator, skeleton templates and example of objects list. Object definitions provided by UNICOS are split into: I/O Objects (Digital Input, Digital Output, Analog Input, Analog Output etc.), Field Objects (OnOff, Analog, Analog Digital, Controller, Alarm etc.) and Process Control Objects (PCO). The objects are used as a common language by process engineers and programmers to define the functional analysis of the process. Each object has unique identifier composed with its equipment type, location and name as well as set of parameters corresponding to its class. Moreover, parameter part contains information about relations between objects forming process logic.

### 2.2 Standard PLC software production procedure

All control system components as well as relationships between sensors and actuators are classified, aggregated and well described in several databases. To extract information necessary for software production from databases the generator of process specification and PLC hardware configuration has been developed at CERN. As a result spreadsheets with a list of objects are obtained. Each worksheet contains information about one type of objects and its parameters in compliance with UNICOS class definition. For example the specification for one PLC contains 5623 objects classified in 16 types.

Beside databases with CDCS components other spreadsheets gather information about process logic. They define system interlocks, alarms, process phases (cool down, nominal operating conditions, warm up), behaviour of valves and heaters depending on functional position in the system (cryogenic circuit) and communication with other systems e.g.: vacuum system, power converters, cryogenic interconnection boxes. According to this information and with the use of skeleton templates provided by UNICOS numerous code templates with the CDCS logic have been created.

All variables together with their PLC and SCADA memory assignment are generated from the object list and code templates using the automatic tool called UNICOS S7 Generator. The same tool performs object list validation (ranges, limits, duplication and discrepancies) and creates code for process logic and PLC--SCADA communication middleware.

Finally a PLC project including hardware configuration corresponding to the code generated in respect to memory addresses is created. After all mentioned above operations UNICOS base line library along with the code already generated can be compiled and deployed on production machine (Fig. 3).

### 2.3 CDCS PLC software production procedure

Large number of elements and complexity of non regular parts of the CDCS forced extension of standard PLC software production procedure used at CERN.

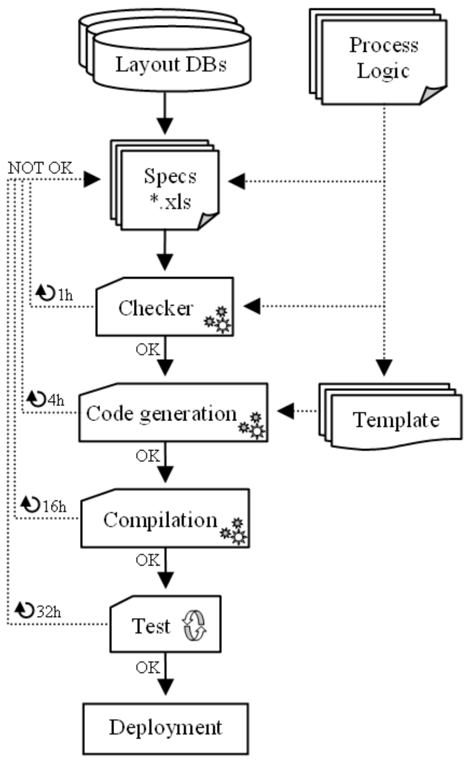


Fig. 3. CDCS software production procedure

Amount of time required to run the whole described above procedure had been around two weeks. The standard validation tools included in UNICOS generator do not cover all specific user defined parameters therefore all of them had to be checked by the programmer before code generation. Moreover, code covering non standard parts of process had to be written manually. As a consequence of significant human participation in code production as well as complicated process structure the large number of mistakes had been discovered. Some of the faults had been detected during the last step - compilation or even worse directly on the machine after deployment. Therefore the CDCS PLC software production process had to be started again from the beginning causing deployment delay and posing real risk of machine damage.

After experience in CDSC software deployment in first tested LHC sector the measures have been taken to avoid hazards described above. The most critical parts have been identified and extended: process specification automatic check, additional generation tools covering non standard parts of code, external functions instead of recurrent code in templates and software test on developer PLC.

#### 2.4.1. Specs validation tools – checker

For automatic validation of list of objects a new tool was developed. The aim of this tool was to shorten the whole code generation process time, by reducing the number of mistakes in specification at the very beginning, to ensure the code is generated correctly and minimize the risk of repetition of the CDCS software generation procedure.

The original list of objects is a spreadsheet, which is not convenient for verification. For maximum efficiency data should to be in the database. To keep internal consistency between data, it was decided to use existing database of S7 Unicos Generator. Our tool is written in PERL script language mainly because of its ease of use while performing complicated operations with files and text. The tool is constantly developed and now covers most of aspects of the project, analyzing the specification for compliance with different rules. What is a fairly simple task for a PERL program was a pretty big challenge for human. Not only one can easily miss something during comparison of long strings of numbers but it also requires a lot of time. The early version of this tool was aimed only on few yet very tedious aspects of the specification, namely the communication variables and logic parameters for various objects. Afterwards more sophisticated checks were applied: object dependencies (e.g.: checking the interlocks for given valves, sensor names and positions for objects as well as default values and ranges), completeness of the data (counting the objects and searching for required sensors) and UNICOS rules (checking if mandatory fields are correct). It was also made compatible with non standard tunnel parts as ACS which requires a totally different set of rules. With the use of process logic and layout databases the tool has become a knowledge base that is easy, and what is most important, fast to use. Using this approach we can easily identify most common errors in specification and clear logs allows for easier corrections.

Using validation tools most errors are discovered within an hour even before the code generation begins which has significantly decreased time necessary for the whole process.

#### 2.4.2. Additional generation tools

Process specifics do not allow the Unicos S7 Generator to generate all necessary code for CDCS. Previously, substantial piece of code responsible for non standard project parts had to be prepared manually by a programmer. This coupled with a large number of objects describing the process caused of a big number of mistakes. In order to eliminate these mistakes a number of additional tools have been developed to substitute the programmer. These tools similarly to the checker are written mainly in PERL language and made on the basis of data from UNICOS Generator.

First, preparation of the code, responsible for calculations of minimal, average and maximal values, has been automated. In total, around 4200 different calculations have been implemented, where single calculation may be derived even from 250 arguments. In the next step, code covering the communication with external systems has been automated. Total number of transmitted variables is up to 5000.

Later, generation of the code, covering the calculation of readiness triggers of the cryogenic system in the scope of circuits powering the superconducting magnets (Cryo Start and Cryo Maintain), has been automated. Due to large variety of system's particular parts, created code has been characterized by little repetitiveness what posed additional difficulty. For example one of the 64 Cryo signals is derived from the state of 250 thermometers, liquid helium level gauges and pressure sensors readouts, strictly specified in the process logic. Manual preparation of the code, covering Cryo Start and Cryo Maintain triggers for one of the LHC sector, had taken 5 days. Currently the time needed has been reduced to a few minutes because of the automation.

In parallel to developing tools replacing the programmer, UNICOS generator capabilities have been extended. Process logic of the cool down of the LHC accelerator defines 12 main phases of work for each of the CDCS parts. Switching between 12 subsequent phases is possible only after the conditions specified in transition are fulfilled and operator has accepted the switch to the next step. These rules had to be implemented in the program. Since standard UNICOS tools did not allow for implementation of such a complex logic an alternative sequencer had to be developed. Our sequencer is split in two parts - first, specific for each Process Control Object, evaluates the transitions (system state). Transitions depend on the subsystem because of the diverse nominal work temperature. Second, common core multiplies the matrices of allowed phase changes definition with current system state. The result of the calculation determines the possible phase changes.

The use of additional code generation tools significantly improved the reliability of code and reduced the time needed to create the application. Now generation of all the code necessary for one sector is possible within four hours.

### 2.4.3. Logic templates and external function

Automatic software generation for CDCS also required creation of several templates. These templates were prepared on the basis of process logic and skeleton templates provided by UNICOS. In the first stage of template creation system objects with similar properties, functionality wise, have been isolated and next grouped. Logic templates were prepared for each group. Finally 136 templates have been created that operate: 69 controllers, 38 valves, 18 heaters. Each of these templates representing actuators has three parts which can be distinguished: header, containing information about alarms, body with logic and footer containing interlocks. In case of templates concerning controllers only the logic part and standard UNICOS environment code fragments can be distinguished. Beside the templates describing the behaviour of single system objects, global templates have been prepared. They are responsible for: machine alarms and interlocks, global logic, sequential control with phase transitions definitions and Process Control Object logic managing the groups of objects.

Templates prepared in this way had been working properly but any process logic modification had required subsequent regeneration and compilation of a group of objects affected by change and therefore substantially impeded the time needed to complete the project. Due to this fact, additional review of object properties has been performed. As a result, all recurrent pieces of code have been singled out and moved to external functions. Therefore, only the standard code from UNICOS environment and external function calls along with their arguments remained in the templates.

Now, using templates with external functions, in the case of modifications of process logic only affected function must be changed. This makes the system easier to maintain and allows flexibility in modifications.

### 2.4.4. Software test on developer PLC

The last step before system deployment on the production machine is software check on the developer machine by the process engineer. Although the developer machine is not connected to the real components in the LHC tunnel it is possible to check all the objects due to UNICOS characteristics allowing for simulation of any object therefore the state of process. In the next steps process engineer checks the interlocks, alarm system, phases and global logic ensuring that the translation of process logic to machine code has been performed correctly. Only positive result of this test allows software engineer to deploy the project on the production machine.

## 3 Conclusions

Automatic approach is necessary to generate the code for such a large scale control system. Currently 4 LHC sectors are being cool down to nominal temperature using software generated according to methodology described in this paper. Two of them are in temperature around 2K. The next 4 sectors and accelerator cavities are ready for cool down and start for all of them have been scheduled for coming days. The main advantages of automatic code generation are: reduced software production time and effort, increased code reliability, minimised risk of human mistakes and simplified long term maintenance.


## Acknowledgement

We would like to thank our colleagues from AGH University of Science and Technology in Kraków for their effort in software production for CDCS. Moreover we would like to thank all members of CERN AT/CRG/IN and AB/CO/IN section for their help, support and contribution.


## References


BRÜNING O., COLLIER P., LEBRUN PH., MYERS S., OSTOJIC R., POOLE J., PROUDLOCK P. ed. 2004, LHC Design Report, Vol 1, CERN-2004-003, CERN Geneva, Switzerland.

The UNified Industrial COntrol System (UNICOS) Framework: http://cern.ch/ab-project-unicos