

## Distributed Execution and Communication in a Process Control Library Application

Oana Rohat <sup>#1</sup>, Dan Popescu <sup>#2</sup>

#1 PhD Student, Politehnica University of Bucharest. +40723872939,  
#2 PhD Professor, Politehnica University of Bucharest. +40766218363,

### ABSTRACT

As automation systems evolve to more complex applications there is a need to improve efficiency in the development time and reducing application errors while maintaining the safety and reliability levels. This is why we designed a process control library of advanced control and optimization algorithms for large – scale industrial plants, written in a standardized format based on IEC 61499. This paper presents the structure, design and functions of this library as well as the communication mechanisms allowing the integration of its components in a local or distributed application.

**Key words:** distributed communication, IEC 61499, process control library, remote execution

**Corresponding Author:** Oana Rohat

### INTRODUCTION

Today the industry uses automation systems that are based more and more on programmable controllers. This makes the design of the control applications more software intensive, thus increasing the number of functions and the level of complexity that can be achieved. Unfortunately an increased complexity leads to lower flexibility and harder application debugging. The new developed IEC 61499 standard gives a possible solution by supporting the design of applications distributed on several controllers.

The IEC 61499 standard is the successor of the IEC61131-3 standard, widely used as a programming language by many PLC manufacturers [1, 2]. The standard provides a great

support in designing modular applications based on function blocks networks, allowing the reconfigurability, portability on hardware platforms from different manufacturers and interoperability of software components [3]. It also allows designing fully distributed applications by implementing the basic generic communication mechanisms and interfaces needed for inter-device communication and application management. These characteristics will be used as a base for increasing the efficiency and lowering the development time of process control applications by designing a process control library of standardized, modular, scalable, reusable algorithms. The library will provide process control engineers algorithms, functions, control sequences, process and communication interfaces that will be ready to implement and use in the control logic of specific applications. Even more, they will have the possibility of executing these algorithms in the library, in a distributed architecture, thus lowering the application complexity and the controller load.

Current process control libraries like Matlab or LabView can be acquired at a high cost and are mostly used in academic research. Their components mostly address areas like modeling, optimization and simulation in the first case, and measurement problems in the second one. A process control library for embedded closed loop control and IPMCS (Industrial measurement and control systems) is presented in [4] but its results and future research are not easy accessible.

The objective of this paper is to show the structure, functions and design of a library of reusable process control algorithms, available online, oriented to the practical implementation and use of the available process control functions in local or distributed configurations. The paper also presents how an engineer can organize the development of an application for an easier integration of the library components and how he can use the library components as a distributed application using specific communication blocks.

Other aspects regarding the library development like storage mechanism and process interfaces development were detailed in [5].

## **LIBRARY OF ALGORITHMS FOR PROCESS CONTROL**

The process control library is a web application that provides access to different control, modeling, optimization algorithms as well as communication and hardware interface functions. The library components will address algorithms and functions from simple to more complex and provide support to process control engineers in developing their applications.

The library components are represented using the recent IEC 61499 standard that supports developing hardware independent function blocks that allow reusability, reconfiguration, developing fully distributed applications using an open representation. This means one can develop specific algorithms in a standardized format independent of the hardware platform on which they will run and using any compliant software application.

The library structure is detailed in Fig. 1. The interaction options depend on the type of user. Four types of users were considered: common users that need no registration, registered users, advanced users that are experienced engineers and library administrators.

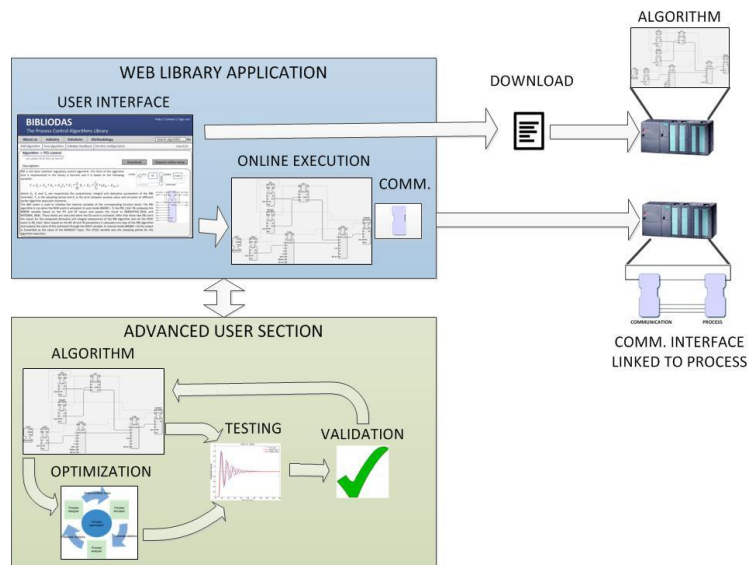


Fig. 1. Web Library structure

A common user can browse different algorithms (basic functions, process or communication interfaces, control sequences and algorithms, safety, modeling or optimization algorithms etc.), read its characteristics, and read the test results from the validation stage, process conditions and feedback from other users. He can use a selected algorithm in an offline configuration by downloading the specific file written in the IEC 61499 standard. The IEC 61499 representation ensures an open platform development so the algorithm files can be used in any IEC 61499 compliant programming software (like FBDK [6], ISaGRAF [7] or nxtControl [8]). By adding dedicated process interface functions, the downloaded algorithm can be loaded on any IEC 61499 compliant controller and can be run without the need of additional changes.

In addition to the options of a common user, a registered user can use an existing algorithm in an online configuration that will allow him to run that algorithm in the library, as detailed in section IV. Registered users can also suggest new algorithms or can add process feedback. These new algorithms will be added to the library only after they are validated by an advanced user.

Advanced user accounts can only be created by the library administrator. They are experienced engineers with access to simulation platforms, process models from different industries, to different mathematical tools for algorithm optimization, testing and validation (like Matlab). They have great responsibility as they must ensure the integrity and correctness of the library components by validating the suggested algorithms.

The validation sequence is comprised of several steps. For an algorithm to be added to the library it must either address a new problem than the existing ones or come with an increased performance solution to an existing problem. Also, the algorithms must conform to a defined methodology and format. The next step is the actual testing of the algorithm to see if applied to a simulated process model it gives the specified results. Based on the results obtained in executing the validation steps the algorithms will be either added to the library or dropped.

Administrators are the librarians that will take care of the library content, users and database management, search optimization.

In developing the library a great attention was given to ensuring the reliability and efficiency of the algorithms so that their correct use can help the process control engineer in developing high performance applications while not affecting system robustness and allowing a high execution speed even for the online execution.

## USING THE LIBRARY COMPONENTS AS PARTS OF A DISTRIBUTED APPLICATION

The IEC 61499 standard defines the system configuration model as a set of devices, having zero or more resources, which run one or several applications using specific communication or process interfaces [9]. An example of a system configuration can be seen in Fig. 2.

Many existing process control applications use a single PLC or DCS controller for the management of a site or a plant. In this case the system configuration includes a single device with one or more resources. The application that contains the algorithm will be run locally on that device.

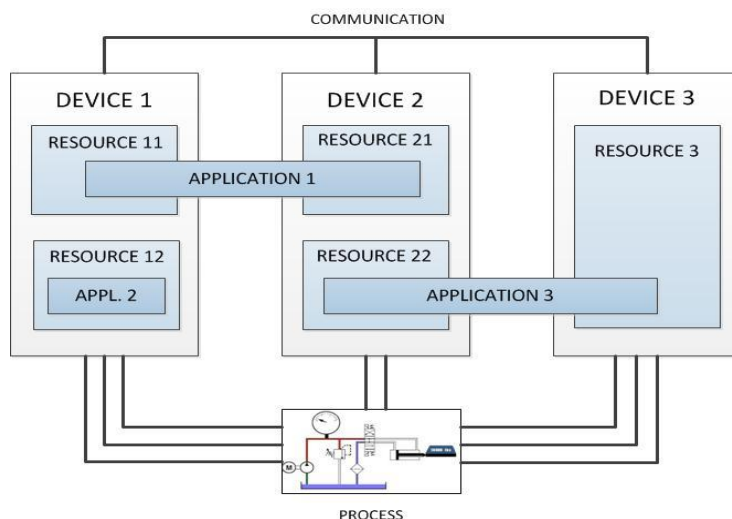


Fig. 2. System configuration

Based on the use of the function blocks of an application, the process control engineer can split that application in several sub-applications. These can be configured to run on several devices based on the corresponding functions. For example, the function blocks for the pump control or the ones implementing the behavior of a HMI panel can be “cut” from the application and assigned to dedicated controllers. In this way the execution of the algorithm will be distributed between the devices involved. By adding communication function blocks in the points where the cutting occurred the exchange of information in the event and data flow will not be perturbed.

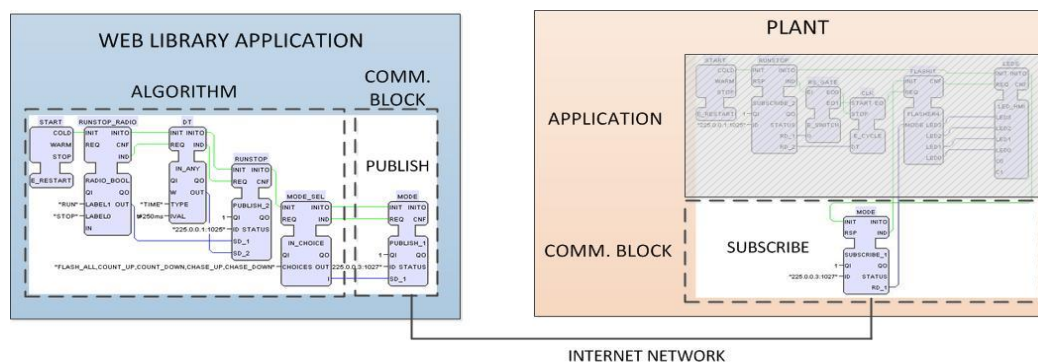
The IEC 61499 standard offers great support in programming and executing completely distributed applications [10, 11]. It allows the process control engineer to imagine and test the

functionality of the application as a whole and then split it on the devices involved. This and the openness of the standard fits with the objectives of the library of providing reusable algorithms that need no or very little additional configuration effort. Even more, this provides the needed support for implementing the option of running the algorithm online. This will be done by configuring the web library server as a remote device running different applications that communicate with different devices from different plants.

When cutting an application for distributed execution communication interfaces must be added at both sides of the cutting points to allow the information flow. The algorithms of the library must have a generic form from the communication point of view. The developers of the library will not know the exact application on which they will run so a standard communication interface will allow receiving inputs and sending commands to any controller implementing the IEC 61499 standard.

The IEC 61499 standard defines communication interface function blocks as a way of interacting with the hardware equipment and even suggests two generic communication patterns: publish/subscribe (uni-directional) and client/server (bi-directional). Starting from these pattern definitions a process engineer can develop function blocks for any industrial communication protocol (like OPC, Modbus, Profibus, CAN etc.). The free FBDK application development software provides basic examples of function blocks implementing the services of the Internet protocol suite [9]. They can be used in creating distributed applications that share information via local or Internet-based communication networks. Commercial development applications like *nxtControl* offers also more complex communication functions, but their implementation is limited as they don't conform strictly to the standard.

The generic communication of the algorithms with the application from a plant in an online execution was illustrated in Fig. 3. The whole plant application appears as a "black-box" for the developers of the web library algorithms. The process control engineer has access to all the information needed regarding the algorithm used so that he can have complete control over the functionality of the whole application. The publish/subscribe blocks provide the easiest communication protocol. They allow sending a set of variables that are set as input of the publish block to the output of the subscribe block. The two blocks communicate over a local or internet network based on an address and port received as input.



## WEB LIBRARY APPLICATION

The library is designed as a web application with a database for storing the algorithms files, a runtime platform based on Java for executing them and an advanced operations platform based on Matlab with several process models used for testing, validating and optimizing the algorithms.

The web library application allows users to browse for algorithms based on the industry for which they apply (Oil & Gas, Chemical, Energy, Water, Manufacturing, Building) or based on the type of algorithm they search for (Basic function, Process interface, Communication interface, Process control, Safety & security, Modeling & optimization), as it is illustrated in Fig. 4. An algorithm can be found in both sections based on its function and characteristics. Each industry section has one or more subsections representing particular installations specific for that industry. For example in the Oil & Gas section the subsections can be Refinery, Warehouse, Bitum factory, Biofuel factory, Burner etc. In each subsection algorithms for different types of equipment can be found. For example, in a refinery controllable equipment are pumps, compressors, regulators, vanes etc. Each of these can have zero or more algorithms, from generic commands to specific control sequences. This way of browsing will be more useful for inexperienced users that don't know exactly the type or name of the algorithm they are looking for.



Fig. 4. Application menu

After selecting an algorithm, the user will be presented its main characteristics, the input and output parameters, test results and feedback from other users (see Fig. 5). He will have the possibility of using the algorithm offline, by downloading it and inserting it in his process control application, or online by configuring the parameters of the communication block. As mentioned before, the online execution is only available for registered users.

After logging in, the application determines the user type and the menu options that are available. A registered user can request an online setup while an advanced user has also access to the Add algorithm, Test algorithm, Validate feedback and On-line configuration options as can be seen in Fig. 5.

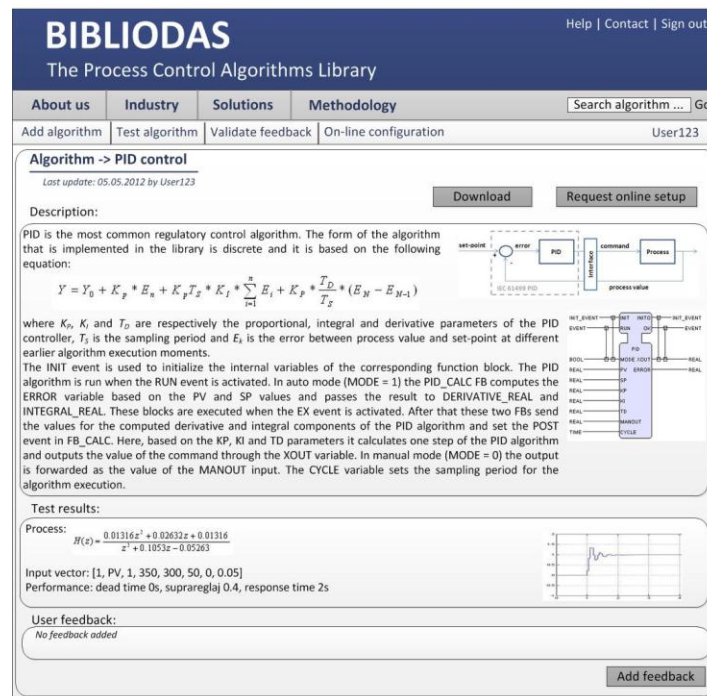


Fig. 5. Algorithm description page example

As stated before, library components can be used offline, by downloading the algorithm file from the library, or online, by configuring the communication settings and using the library server as a remote device.

Based on the algorithm type selected, in an offline execution the user receives the required function block, interface or system configuration representing a control sequence or an advanced algorithm for modeling or optimization. The user can include these function blocks of system configuration files in the existing control logic of a plant and, based on the resources available and the system architecture, he will decide whether the algorithms will be run locally (on a single controller) or will be distributed across several devices and/or resources. That being said, downloading an offline algorithm is similar to accessing the library of function blocks or sample programs of the developing environment, except the web library will cover more and sometimes more complex algorithms that are better documented and can be used in any of the existing IEC 61499 programming applications.

The online execution means that the algorithm is executed online, in real time, on the library server or in a browser applet, based on a specified set of input parameters, and only the result is sent to the user. By configuring an online execution of an algorithm the user creates a remote device with a specific communication interface. The user will have the possibility to select the desired communication protocol from the ones available and will have to provide the communication server address from the device and/or the address of the input parameters according to the algorithm and the communication protocol selected.

This option can be requested only by the registered users so that the advanced users can have a better control on the number of executed threads and performances. By accessing the online configuration menu, an advanced user can see the current executing algorithms, the ones that

are pending, the users that requested them, the network addresses, the communication protocol, the start and end date of the execution. He can also start or stop an algorithm from executing.

The online execution will be available for control algorithms and sequences, for safety, modeling and optimization algorithms. The more simple components of the library (like basic function block, process or communication interfaces) will not be available for online configuration since there is no practical need for such applications.

## CONCLUSION

This paper presents a solution for a process control library of reusable algorithms written in a standardized format based on IEC 61499 that allows portability on equipment from different vendors and on different development environments that conform to the standard. We proposed a web library structure and identified the characteristics of the IEC 61499 standard that support the use of the library components by creating sub-applications and using them either local or in distributed system configurations. These helped us design the web library application so that all the functional its needs are accomplished.

Future work will include testing the limitations of the library in executing the algorithms online, implementing communication security mechanisms and developing complex algorithms in the IEC 61499 standard.

## REFERENCE

- [1] R. W. Lewis, Modelling control systems using IEC 61499: applying function blocks to distributed systems, in *Control Engineering Series 59*, IEE, U.K., 2001.
- [2] V. Vyatkin, IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review, *IEEE Transactions On Industrial Informatics*, vol.7, issue 4, 2011, p. 768 – 781.
- [3] IEC 61499 standard. Part 3 Application Guidelines: Function blocks for Industrial Process Measurement and Control Systems. Technical report. Geneva, 2005.
- [4] T. Strasser, F. Auinger, A. Zoitl, Development, implementation and use of an IEC 61499 function block library for embedded closed loop control, *2nd IEEE International Conference on Industrial Informatics INDIN*, 2004, p.594 – 599.
- [5] L. Ocheana, O. Rohat, D. Popescu, G. Florea, Library of Reusable Algorithms for Internet - Based Diagnose and Control System. *14th IFAC Symposium on information Control Problems in Manufacturing*, vol. 14, part 1, 2012.
- [6] Holobloc Inc., Resources for the New Generation of Automation and Control, Function Block Development Kit (FBDK). Available: [www.holobloc.com](http://www.holobloc.com).
- [7] ISaGRAF, ISaGRAF Workbench. Demo available: [www.isagraf.com](http://www.isagraf.com).

- [8] nxtControl, nxtStudio. Demo available: [www.nxtcontrol.com/no\\_cache/en/support/downloadarea.html](http://www.nxtcontrol.com/no_cache/en/support/downloadarea.html)
- [9] V. Vyatkin, IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, NC, USA, 2012, p. 75 – 113.
- [10] T. Bezák, Usage of IEC 61131 and IEC 61499 Standards for Creating Distributed Control Systems. Publ. Univ. Verlag, 2012.
- [11] Y. Wei, Implementation of IEC61499 Distributed Function Block Architecture for Industrial Measurement and Control Systems (IPMCS). *Total Enterprise Solutions Conference, ICAM*, 2001.