Computers in Industry 72 (2015) 47–67

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

An architecture to integrate IEC 61131-3 systems in an IEC 61499 distributed solution

Stefano Campanelli [b], Pierfrancesco Foglia [a,*], Cosimo Antonio Prete [a]

[a] Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Via Diotisalvi 2, 56126 Pisa, Italy
[b] ION Trading, Via Ceci 152, Pisa, Italy

ARTICLE INFO

Article history:
Received 30 May 2014
Received in revised form 14 April 2015
Accepted 22 April 2015
Available online 30 May 2015

Keywords:
PLC
SW reuse and integration
Distributed control systems
IEC 61131
IEC 61499
Discrete manufacturing
Manufacturing automation

ABSTRACT

The IEC 61499 standard has been developed to allow the modeling and design of distributed control systems, providing advanced concepts of software engineering (such as abstraction and encapsulation) to the world of control engineering. The introduction of this standard in already existing control environments poses challenges, since programs written using the widespread IEC 61131-3 programming standard cannot be directly executed in a fully IEC 61499 environment without reengineering effort. In order to solve this problem, this paper presents an architecture to integrate modules of the two standards, allowing the exploitation of the benefits of both. The proposed architecture is based on the coexistence of control software of the two standards. Modules written in one standard interact with some particular interfaces that encapsulate functionalities and information to be exchanged with the other standard. In particular, the architecture permits to utilize available run-times without modification, it allows the reuse of software modules, and it utilizes existing features of the standards. A methodology to integrate IEC 61131-3 modules in an IEC 61499 distributed solution based on such architecture is also developed, and it is described via a case study to prove feasibility and benefits. Experimental results demonstrate that the proposed solution does not add substantial load or delays to the system when compared to an IEC 61131-3 based solution. By acting on task period, it can achieve performances similar to an IEC 61499 solution.

1. Introduction

Programmable Logic Controllers (PLCs) are some of the most widespread devices in industrial automation, being used in industry for decades [30]. In 1993, the International Electrotechnical Commission (IEC) published the IEC 61131-3 standard [1] in order to define a common programming interface for PLCs produced by different manufacturers. Since then, the standard has been widely adopted among PLC producers.

Evolution of computer networks brought the technology to realize control applications distributed among different devices. The new market demands [32] for flexibility and reconfigurability in manufacturing and process industries motivate the need of a transition from centralized to distributed control systems [2,4,31,41]. Distributed control is highly desirable in manufacturing industry, however, the design of a distributed control system is more challenging than a centralized one [4]. In particular, in an IEC 61131-3 environment, the aspects of distribution of control logic (such as the mapping of the code modules or variables to the devices and the communications) must be considered from the beginning of the design phase [40]. This complicates the design of the distributed control system and, moreover, it limits the ability to reconfigure the system, since changing the control system environment requires a redesign effort and the modification of existing software modules [40].

In order to facilitate the design of distributed control systems, IEC proposed the IEC 61499 standard [3]. The standard introduces in the development of distributed control application advanced concept derived from distributed systems and the software engineering discipline, such as encapsulation, independence of the control logic from the communication infrastructure, and development of software component independently from their mapping to hardware. The standard defines an open architecture, based on an event driven execution approach, that permits to model and design control applications organized in software components distributed across networked devices [4]. Practical case studies [5,6,7] have proved the benefits of the standard, and

* Corresponding author. Tel.: 0039 0502217530.
E-mail addresses: ste.campanelli@gmail.com (S. Campanelli), foglia@iet.unipi.it (P. Foglia), prete@iet.unipi.it (C.A. Prete).

http://dx.doi.org/10.1016/j.compind.2015.04.002
0166-3615/

there are some commercial tools supporting it already on the market (ISaGRAF [8] and nxtControl [9]).

Since IEC 61131-3 has been in use for years [30], there are both a large number of control systems in operation and already developed and tested software libraries that are based on this standard. Unfortunately, it is not possible to directly execute IEC 61131-3 applications in an IEC 61499 based runtime environment [4,11], so the adoption of the IEC 61499 would result in the waste of such existing code, know-how and competences, as also noted for the batch process industry and the IEC 61512 standard [33]. Besides, the adoption of IEC 61499 in an existing system would result at least in the modification of the software implementation of the control algorithm, with consequent need of new design and testing phases and interventions on the production line with the eventual stop of operation. The switching cost to the new standard is then perceived to be very high, and it could be lowered if some of the IEC 61131-3 investments are retained [33]. For these reasons, companies that have been already slow to adopt languages other than IEC 61131-3 [30], are even less inclined to convert existing systems to IEC 61499 [31], nevertheless, many of these systems still may take advantage from distribution of control logic to increase interoperability, reduce human work, improve control decisions, reliability, scalability, reconfiguration and deployment [2,4,31,41]. One solution to minimize the cost, retain the investment and exploit distribution consists in developing some components in the new standard, while keeping the other software modules already written in the old.

For instance, starting from an existing IEC 61131-3 system, there is the need to redesign new components to allow a more flexible distributed control when the application is running on more devices. In such a situation, it is convenient to adopt, for the new components, the IEC 61499 standard, while reusing, to retain the investment, the available IEC 61131-3 code. In a different use case, starting from an IEC 61131-3 system, the introduction of new hardware components determine the introduction of new software modules devoted to their management. Such modules can be written again in the new standard, to exploits the benefit of it. As a general observation on the advantage of coexistence of both the standards, we can say that the IEC 61131-3 standards permits to see software modules as hardware-like modules (ladder diagrams or FB diagrams), while the IEC 61499 standard gives the system a more software oriented vision, with components, connections and events. The IEC 61499 vision is useful for system designers, who are usually experienced software/control engineers and have to deal with the distributed aspects of control. Nevertheless, the hardware-like vision is useful for plant staff, which, although inexperienced in advanced control engineering techniques, need to keep the plant operational [11,30].

Such examples motivate the needs of architectures that permits the coexistence of code modules written in the two standards [12]. Such coexistence is not simple to realize by system designers, as IEC 61131-3 modules can't be directly executed on IEC 61499 runtime [4,11].

In this paper, we deal with such problem, and we propose an architecture, namely LE-INT (LowEffort-INTegration architecture) to realize the coexistence of code modules written in the two standards without requiring, for system designers, effort and knowledge higher to the ones required to write software modules in the two standards. In particular, the proposed solution does not require modifications to the run-time support of the two standards, as well as modification to the development tools, as it utilizes defined features of the two standards.

In the proposed solution, there are physical devices able to execute IEC 61131-3 and IEC 61499 code modules. Such modules can be on the same device or on different physical devices; they can exchange information through appropriate mechanisms that encapsulates the interaction and the implementation details of the communications. As a consequence, IEC 61131-3 developers and staff people do not need to know the details of IEC 61499 code, that is abstracted as IEC 61131-5 function blocks, with no loss of acquired competence and development effort. From the IEC 61499 perspective, the IEC 61131-3 control logic is abstracted as one or more Service Interaction Function Blocks (SIFB), that can be utilized as standard IEC 61499 modules, adhering to the concept of encapsulation of the standard.

Based on LE-INT, we define a methodology for the integration of IEC 61131-3 code modules in IEC 61499 logic. In this way, as we will show with a case study, existing IEC 61131-3 code can be kept and easily integrated in an IEC 61499 distributed solution.

Main fields of application of LE-INT are the manufacturing and automation processes (such as, but not only, discrete part production lines [37]) organized in a modular way and as sequences of processing steps. Such systems may have a not sequential information flow and may include branching paths. Examples can be found in [37,38,39]. LE-INT is effective in these systems, to realize the following use scenarios (use cases):

(1) transition to distributed mixed systems using both standards from existing IEC 61131-3-based systems;
(2) insertion of one or more existing IEC 61131-3 system in an existing distributed IEC 61499 system;

In the first use-case, LE-INT can be used to transform a centralized or several independent IEC 61131-3 systems in a distributed system based on both IEC 61131-3 and IEC 61499 standards. In this scenario, LE-INT permits to use the distribution features of IEC 61499, while reusing existing IEC 61131-3 code. A limit of our approach consists in the need of performing a decomposition of the existing system (in case of a centralized system) that is not straightforward and requires effort. However, when the manufacturing process is organized in distinct steps [38] or the system is composed of different physical subsystems [39], usually the centralized system is structured in a modular way where different code modules manage different parts of the physical system, communicating via global variables [39]. In these cases, the effort of decomposing the system is greatly reduced, since code is already decoupled.

The second use-case refers to the insertion of an existing IEC 61131-3 module in a modular distributed production-line programmed in IEC 61499, as shown in [39]. In this case, LE-INT allows to see, from the IEC 61499 point of view, the IEC 61131-3 system as a software component (a SIFB) that can become part of the distributed application. The disadvantage of this approach consists mainly in the delay added by the mixing of cyclic and event-driven execution models, delay that we will evaluate in the paper, with reference to a case study.

Results of the evaluation indicate that LE-INT performs similarly to a pure IEC 61131-3 solutions and, by acting on task period, it can achieve performance similar to a pure 61499 solution, while the overhead in terms of CPU and memory usage, introduced by the contemporary execution of two runtimes, in negligible. As a summary, the advantages introduced by LE-INT (derived from the opportunity of reuse, distribution and expansion) are not overwhelmed by performance drawbacks.

The rest of the paper is structured as follows: Section 2 presents related works, basic concepts of IEC 61131-3 and IEC 61499 are discussed in Section 3, the proposed architecture is described in Section 4, in Section 5 a methodology of integration is presented via a case study, while Section 6 discusses implementation and performance evaluation. Section 7 concludes the paper.