# Software for Next Generation Automation and Control

Julien Chouinard and Robert W. Brennan, *Member, IEEE*

*Abstract*— **The current trend in automation and control research has been toward "intelligent" systems that are intended to adapt quickly to change while providing extensibility through a modular, distributed design. The key to integration of these systems lies in their control software; however the main barrier to flexibility lies in the difficulties associated with the development of this software. In this paper, we look at the basic requirements for this next generation of automation and control systems and introduce a new software tool that is based on the IEC 61499 standard that is intended to address the issue of software development.**

## I. Introduction

In order to maintain competitiveness in today's global market, industry requires systems that are capable of quickly responding to change while maintaining stable and efficient operation. Increasingly, the industrial control system is viewed as being central to achieving this goal.

The main barriers to success in this area however, result from the combination of increasingly stringent customer requirements (e.g., high quality, customizable, low-cost products that can be delivered quickly) and inherent system complexity (i.e., these systems are, by nature, distributed, concurrent and stochastic). Although technology has become increasingly sophisticated to deal with these issues (e.g., through advanced robotics and computer numerical control), without adequate control, the result is often a collection of "islands of automation" that lack the necessary integration for truly responsive behavior. As a result, new control software and hardware approaches are required to realize a system that is flexible (i.e., capable of reconfiguration) and responsive (i.e., capable of recovering from disturbances).

In this paper, we focus on a new software model, IEC 61499 [1], that is intended reduce the barriers imposed on the flexibility and responsiveness of industrial automation systems by current control technology. The paper begins with more details on the requirements for the next generation of automation and control systems. Next, we provide an overview of the IEC 61499 model and summarize the current state of research into the application of this model to industrial control problems. Although a considerable amount of work has already been conducted in this area, there has been a lack of industrial strength software tools available for IEC 61499 developers. In section 4, we focus on a new IEC 61499 development environment that can be used to apply this approach to real industrial automation and control problems. Finally, the paper concludes with a brief summary and outlook to the future of these systems.

Manuscript received March 31, 2006.

J. Chouinard is with ICS Triplex ISaGRAF Inc., 9975 Catania Avenue, Suite U, Brossard, Quebec, J4Z 3V6 Canada (e-mail: jchouinard@icstriplex.ca).

R.W. Brennan, is with Schulich School of Engineering, University of Calgary, 2500 University Dr. N.W., Calgary, Alberta, T2N 1N4 Canada (phone: 403-220-5798; fax: 403-282-8406; e-mail: rbrennan@ucalgary.ca).

## II. The Requirements for the Next Generation of Automation and Control

As noted previously, the key to competitiveness in today's global market is the ability to respond quickly to change while maintaining stable system operation and efficient use of available resources. In the manufacturing domain, there has been a considerable amount of interest recently in distributed intelligent control solutions to address this issue. In particular, research in this area has moved away from traditional, monolithic, centralized solutions, towards distributed approaches where the architecture of decision-makers ranges from hierarchical to non-hierarchical (or "heterarchical") [2].

Distributed intelligent control involves matching the control model more closely with the physical system. This is particularly relevant to manufacturing control systems that are required to control widely distributed devices in an environment that is prone to disruptions. With this model, control is achieved by the emergent behavior of many simple, autonomous and co-operative entities (i.e., agents) that "decide locally not only how to act (as subroutines do), and what actions to take (as objects do), but also when to initiate their own activity" [3]. The natural fit of multi-agent systems technology to manufacturing problems has resulted in many applications in this domain [4]. Consequently, this has led to Holonic Manufacturing Systems (HMS), a manufacturing-specific application of the broader MAS approach [5].

In the manufacturing domain, holonic systems and multi-agent systems are closely related. In particular, a multi-agent manufacturing system consists of cooperative and autonomous manufacturing units, but unlike an HMS, a MAS can be considered as embedding a "general software technology that was motivated by fundamental research questions" [6].

Research in MAS however has played a key role in the development of holonic manufacturing systems. For example, holonic system design is typically in the form of an agent-oriented architecture (e.g., [6]). Because of the close relationship between holonic systems and MAS concepts, object-oriented and agent-based techniques have played an important role in holonic systems research.

Recently, there has been considerable interest in extending this work from the upper, planning and scheduling level of control [7], to the physical device level. For example, members of the HMS consortium focused on defining a low-level control architecture [8; 9] that is based on the IEC 61499 function block standard [1]. With this work as a basis, IEC 61499 function blocks have been used at the device level for dynamic reconfiguration [10; 11], safety management [12], and system validation [13]. This has led to range of industrial applications of real time distributed control such as DaimlerChrysler's holonic control implementation for engine assembly [14], Rockwell Automation's design of a reconfigurable ship-based chilled-water system [15], the integration of RFID (radio frequency identification) systems [16], and recent work on the use of holonic systems for military applications such as sensor management [17].

When agent technology is applied to physical devices, system design and analysis are of utmost concern given real time and safety issues at this level of control. For example, as devices such as controllers, sensors and actuators become "smarter", safety functions that were previously performed by mechanical or electrical interlocks are assumed by computer software that may reside in a single device or might be distributed across multiple devices. Failures in these systems can have a significant impact on a company's bottom line (e.g., profit reduction due to temporary loss of equipment resulting in production downtime), and more importantly, on the lives of people. Because of the more stringent requirements for latency, reliability and availability, it follows that the step from the non-real time or soft real time domain is a large one, requiring new models and methodologies for distributed control.

In the next section we look at a new standard for industrial process control that is intended to address theses issues.

## III. The IEC 61499 Standard

Much of the recent work on distributed intelligent control at the device level has relied on the IEC 61499 standard. This standard was developed by the International Electro-technical Commission to addresses the need for modular software that can be used for distributed industrial process control [1]. In particular, IEC 61499 builds on the function block portion of the IEC 61131-3 standard for programmable logic controller (PLC) languages [18] and extends the function block language to meet more adequately the requirements of distributed control in a format that is independent of implementation.

In developing control applications with this model, the IEC 61499 function block can be thought of in terms of an "enhanced" object. Like recent object-oriented and agent-based models for manufacturing system control, the IEC 61499 function block shares many of the characteristics of the traditional objects and agents used to develop these applications (e.g., a traditional object focuses on data abstraction, encapsulation, modularity, and inheritance).

The function block is enhanced through its recognition of two very specific kinds of messages: data messages (which one would expect of a traditional object) and event messages (which are used to schedule the execution of an algorithm). However, it should be noted here that, from the semantic point of view neither data nor events are messages (i.e., function blocks use the ISO 2382 definition of a message: "an ordered series of characters intended to convey information"[1]). The occurrence of an event, though, can be used to trigger the transmission of a message in a distributed system. The resulting focus on process abstraction and synchronization makes this approach particularly suitable for the control of an environment that is concurrent, asynchronous and distributed.

Current PLC systems typically combine a control application in a single PLC (programmable logic controller) with remote I/O (input/output) devices. With the continuing trend towards small, inexpensive programmable controllers, the possibility of distributed intelligent devices with local I/O is becoming more attractive. As is illustrated in Figure 1, given a programming language that supports this type of distributed application, this can be achieved by the distribution of functional units over a network of communicating processors.

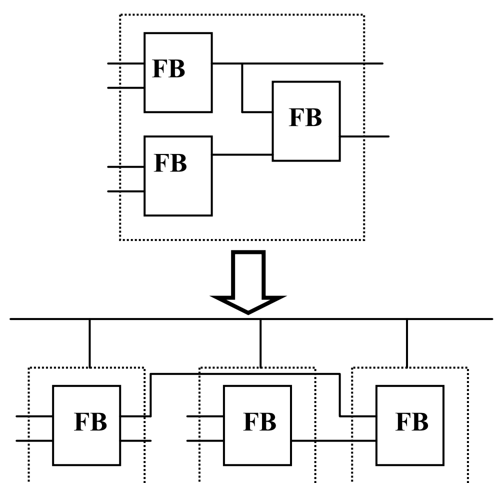

Fig. 1. IEC 61499 distributed applications

One of the main limitations of existing approaches to PLC programming (e.g., ladder logic, IEC 61131-3) is that program execution is based on the conventional PLC scan cycle. This leads to considerable inflexibility in the scheduling of events, which can result in serious difficulties in meeting the requirements of hard real-time control.

As is shown in Figure 2, through a combination of the object concept (e.g., the IEC 61131-3 "function block") and the concept of state diagrams (e.g., the IEC 61131-3 "sequential function chart" or the French GRAFCET language) a solution can be found. As indicated at the bottom of Figure 2, both scheduling of events (i.e., via function block event inputs and outputs) and data flow (i.e., via the typical function block connections) can be specified in a single notation.

This results in the "enhanced" object or function block that was mentioned previously. The shaded portions of the function blocks shown in Figure 2 are responsible for execution control and the non-shaded portions are responsible for functionality (e.g., algorithms and data). IEC 61499 extends the IEC 61131-3 standard through these two basic concepts of functional distribution and event-based execution control.

A detailed description of the IEC 61499 standard and its associated models (i.e., function block, application, resource, device, system) is beyond the scope of this paper. However, the reader may consult [19] for a comprehensive overview of these models and [1] for the detailed standard.

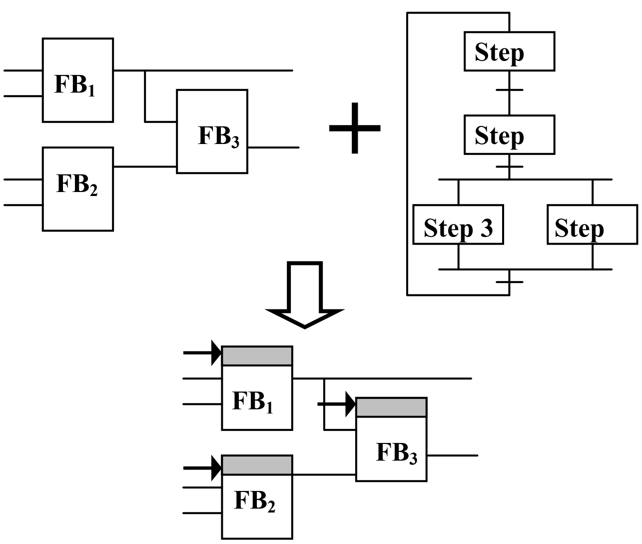


Fig. 2. IEC 61499 function blocks

As noted previously, researchers have been applying IEC 61499 to distributed intelligent control problems for a number of years now. Despite the strong interest in IEC 61499 from the research community however, there has been a lack of software tools for industrial automation and control developers. In particular, extant software tools have been focused on the research community for simulation and testing and are, arguably, not suitable for industrial applications.

The most widely-used research tool for IEC 61499 application development and testing is Function Block Development Kit (FBDK), which was originally developed by Rockwell Automation and is now managed by Holoboc Inc. [18]. This is a Java-based software development that is freely-distributed for research and education purposes. The tool is primarily intended to allow users to develop and simulate IEC 61499 applications on a single platform (e.g., PC, Linux), however there is also a run-time version available (FBRT) that allows users to execute their IEC 61499 applications on embedded Java controllers such as the Tiny InterNet Interface (TINI) [21], Systronix aJile Euroboard (SaJe) [22] and the Simple Network Application Platform (SNAP) [23] platforms. Work is currently underway on an open-source Java-based Integrated Development Environment (IDE) for the development of IEC 61499-compliant library elements for industrial automation and control. This project is currently managed through SourceForge as the O$^3$NEIDA Workbench project [24].

In addition to FBDK, there have been various other research-oriented IEC 61499 software development environments. For example Wang et al. [25] developed a simple function block editor called ICS Developer for use with a distributed real-time operating system, Brennan et al. [26] developed a Java-based tool for dynamic reconfiguration experimentation, and Tranoris and Thramboulidis [27] have developed an engineering support system that extends the IEC 61499 model to software design through the use of the UML (Unified Modeling Language). These have tended to be very specific to individual research projects however, and the majority of recent non-FBDK environments have been extensions or add-ons to FBDK such as the verification work by Vyatkin and Hanisch [28].

Given the maturity of IEC 61499 research, the time is now right for an industry-oriented IEC 61499 software development environment that can be used by control engineers to implement distributed control at the device level. In the next section, we describe a new software development environment that has been developed to enable industrial practitioners to design and develop distributed control applications.

## IV. THE ISAGRAF IEC 61499 SOFTWARE DEVELOPMENT ENVIRONMENT

In order to provide industry with open, flexible and quality automation software technology that allows control engineers to accelerate application development, speed time-to-market and create competitive differentiation, ICS Triplex [29] recently introduced version 5.0 of the popular ISaGRAF software development tool. ISaGRAF version 5.0 extends earlier versions of ISaGRAF to support the design of automation controllers and devices that meet the IEC 61131-3 and IEC 61499 standards.

As noted in the previous section, until now there has been a gap between IEC 61499 research and application: this is primarily because of the lack of industrial IEC 61499 software development tools. ISaGRAF 5.0 builds on years of IEC 61499 development, providing users with the ability to build traditional control systems where the interactions between devices will be regulated automatically and synchronized by the IEC 61499 function block diagrams rather than through the use of manually implemented algorithms.

Figures 3 and 4 provide an example of this process. Figure 3 shows the IEC 61499 device view where the designer makes decisions about where each part of the distributed control application is scheduled and executed. Figure 4 shows the IEC 61499 application view, or the function block application itself. Here the designer creates the specific solution to the control problem using IEC 61499 function blocks.

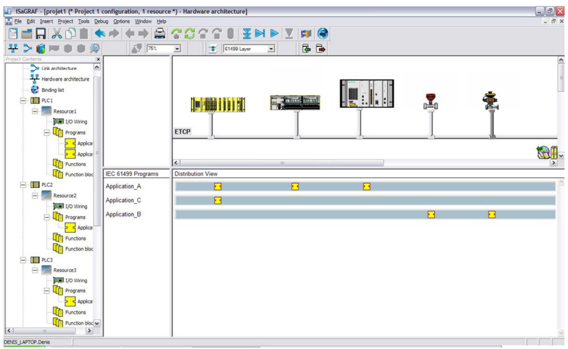

Fig. 3. ISaGRAF IEC 61499 device view.

By providing the designer access to multiple views of the distributed application, ISaGRAF 5.0 makes the development and maintenance process practical and easy. Given the IEC 61499 standard's hierarchical organization of industrial process measurement and control systems (i.e., from the system model down to the function block model), designers can work at various levels of abstraction within ISaGRAF 5.0 to prototype rapidly and to create distributed control applications.

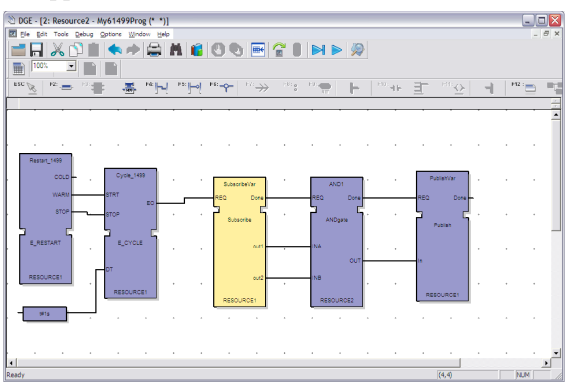

Fig. 4. ISaGRAF IEC 61499 application view.

When developing applications like the one shown in Figure 4, ISaGRAF allows the designer to use the full suite of IEC 61131-3 languages to specify the behavior of individual function blocks. For example, control algorithms may be specified in structured text (ST) or ladder diagrams (LD). This is taken one step further for execution control of function blocks. In order to provide users with an easy transition from traditional IEC 61131-3 systems to new IEC 61499 systems, the ISaGRAF developers chose to allow IEC 61499 execution control charts (ECC) to be specified in the familiar IEC 61131-3 sequential function chart (SFC) notation rather than in IEC 61499 state machines. From a programming syntax point of view, this is no different than the formal IEC 61499 specification. However, from a programming practice point of view, this enhances the usability of the ISaGRAF tool by providing control engineers with a smooth transition from the familiar IEC 61131-3 to distributed control system design and development.

Once the application is developed within ISaGRAF 5.0, it can then be compiled into Target Independent Code (TIC) for execution on the target platform. The philosophy behind TIC is to provide the user with target hardware flexibility. More specifically, the TIC technology enables the control engine to be implemented on any hardware platform (embedded, Motorola, Intel, ARM etc.) and on any operating system (Win XP, CE, 2000, Linux, VxWorks, QNX etc.). TIC can also be used to build automation devices such as tiny controllers, RTUs, PLCs, DCSs and safety systems. In fact, in using ISaGRAF 5.0, any field device manufacturer will have the ability to turn their equipment into an IEC 61499 product. For example, intelligent flow meters can act as a regulator and control a valve while being part of an integrated distributed control system.

With any control application development project, it is useful to be able to test the application before committing it to factor floor. This is particularly relevant for distributed applications, since the system designer will not only need to verify the execution of the application on individual controllers, but also ensure that the application is properly synchronized between distributed controllers.

To aid in the verification process, the application can be simulated directly on the developer's computer using ISaGRAF 5.0. As well, the developer can take this one step further using the ISaGRAF 5.0 demo kit, shown in Figures 5, in conjunction with the development environment. The demo kit includes three embedded controllers that are each equipped with the ISaGRAF 5.0 runtime and an I/O driver. The three controllers are connected via Ethernet to a wireless hub that can be accessed by a workstation running ISaGRAF 5.0.

Fig. 5. ISaGRAF 5.0 demo kit.

Of course, the ISaGRAF 5.0 demo kit also works as an excellent tool for IEC 61499 training. Although general industrial control and PLC training is quite common, there is clearly a lack of training opportunities for distributed control application development. Tools like the ISaGRAF 5.0 demo kit can fill this void and help to train a new generation of control engineers who can use a distributed approach to solving complex industrial control problems.

## V. SUMMARY

In this paper, we have provided an overview of the current approaches to implementing distributed intelligent control at the device level. Although most of the work on software tools at this level has focused on testing environments for research purposes, new industry-oriented tools are becoming available such as ICS Triplex's ISaGRAF 5.0.

Clearly, these new software tools are of utmost importance if IEC 61499 technology is to become a viable option for industry. Without adequate tools to enable practicing engineers to develop IEC 61499 distributed control applications and test them on a variety of targets, IEC 61499 will not realize its potential. More importantly, industry will still struggle to develop truly flexible and responsive systems.

Based on our review of the current work in this area, it is our opinion that research-based tools such as FBDK and industry-oriented tools such as ISaGRAF 5.0 will complement each other. While the ideal of distributed intelligent control at the device level is still not completely realized: tools such as FBDK and initiatives such as the $O^3$NEIDA Workbench will continue to play a strong role in their support of research in this area. For IEC 61499 to be accepted by industry however, IEC 61499 must be implementable in practice. This is best realized with familiar and widely accepted tools such as ISaGRAF. Given this marriage of research and practice, the future of intelligent automation and control systems looks very promising.

## REFERENCES


[1] IEC TC65/WG6, Voting Draft – Publicly Available Specification: Function Blocks for Industrial Process-measurement and Control Systems, Part 1, International Electrotechnical Commission, 2000.

[2] D. Dilts, N. Boyd, and H. Whorms, The evolution of control architectures for automated manufacturing systems, Journal of Manufacturing Systems, 10(1), 1991, 70-93.

[3] Parunak, H. "Autonomous agent architectures: a non-technical introduction," Industrial Technology Institute Report, 1993.

[4] Shen, W., and D. Norrie, "Agent-based systems for intelligent manufacturing: a state-of-the-art survey," Knowledge and Information Systems, 1, 129-156, 1999.

[5] Valckenaers, P., H. Van Brussel, L. Bongaerts, and J. Wyns, "Holonic manufacturing systems," Integrated Computer Aided Engineering, 4, 191-201, 1997.

[6] Bussmann, S. "An agent-oriented architecture for holonic manufacturing control," *Proceedings of the First Open Workshop on Intelligent Manufacturing Systems Europe*, pp. 1-12, 1998.

[7] McFarlane, D. and S. Bussmann, "Developments in holonic production planning and control," International Journal of Production Planning and Control, 11(6), pp. 522-536, 2000.

[8] S. Balasubramanian, R.W. Brennan, and D.H. Norrie, "An architecture for metamorphic control of holonic manufacturing systems," Computers in Industry, 46(1), pp. 13-31, 2001.

[9] Christensen, J. "HMS/FB architecture and its implementation," In: M. Deen (Ed.) Agent-based Manufacturing: Advances in the Holonic Approach, pp. 53-88, 2003.

[10] S. Olsen, J. Wang, A. Ramirez-Serrano, R.W. Brennan, "Contingencies-based reconfiguration of distributed factory automation," Robotics and Computer-integrated Manufacturing, 21(4-5), pp. 379-390, 2005.

[11] Strasser, T., A. Zoitl, F. Auinger and C. Sunder, "Towards engineering methods for reconfiguration of distributed real-time control systems based on the reference model IEC 61499," In: V. Marik, R.W. Brennan, M. Pechoucek (Eds.), Holonic and Multi-agent Systems for Manufacturing, Lecture Notes in Computer Science, Vol. 3593, pp. 165-175, 2005.

[12] J.J. Scarlett, R.W. Brennan, D.H. Norrie, "A proposed high-integrity communication protocol for real-time control," Proceedings of the IMS International Forum on Global Challenges in Manufacturing, Cernobbio, Italy, pp. 759-766, 17-19 May, 2004.

[13] Vyatkin, V. and H. Hanisch, "Component design and formal validation of SFA systems: a case study," 5th IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, Cancun, Mexico, pp. 313-322, 25-27 September, 2002.

[14] Bussmann, S. and J. Sieverding, "Holonic control of and engine assembly plant: an industrial evaluation," Proceedings of the 2001 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, 2001.

[15] Maturana, F., R. Staron, P. Tichy, P. Slechta, and P. Vrba, "A strategy to implement and validate industrial applications of holonic systems," In: V. Marik, R.W. Brennan, M. Pechoucek (Eds.), Holonic and Multi-agent Systems for Manufacturing, Lecture Notes in Computer Science, Vol. 3593, pp. 111-120, 2005.

[16] Vrba, P., F. Macurek, and V. Marik, "Using radio frequency identification in agent-based manufacturing control systems," In: V. Marik, R.W. Brennan, M. Pechoucek (Eds.), Holonic and Multi-agent Systems for Manufacturing, Lecture Notes in Computer Science, Vol. 3593, pp. 176-187, 2005.

[17] P. McGuire, G. Liggins, A. Benaskeur, R.W. Brennan, P. Wojcik, "The application of holonic control to tactical sensor management", IEEE Workshop on Distributed Intelligent Systems, 2006 (submitted).

[18] R. Lewis, Programming Industrial Control Systems using IEC 1131-3 (IEE Control Engineering Series 50, 1996).

[19] R. Lewis, Modelling Distributed Control Systems Using IEC 61499 (IEE Control Engineering Series 50, 2001).

[20] Holobloc, Web Site, http://www.holobloc.com, 2006.

[21] D. Loomis, The TINI Specification and Developer's Guide, Pearson, 2001.

[22] SaJe, "Real Time Native Execution", http://www.systronicx.com/saje/index.html, 2006.

[23] Imsys, "Embedded Java Platform for Network Telematic Applications," http://www.imsys.se/, 2006.

[24] $O^3$NEIDA Workbench, http://sourceforge.net/projects/oooneida-wb/, 2006.

[25] L. Wang, R.W. Brennan, S. Balasubramanian, and D.H. Norrie, "Realizing holonic control with function blocks," Integrated Computer-Aided Engineering, 8(1), pp. 81-93, 2001.

[26] R.W. Brennan, X. Zhang, Y. Xu, and D.H. Norrie, "A reconfigurable concurrent function block model and its implementation in real-time Java," Integrated Computer-Aided Engineering, 9(3), pp. 263-279, 2002.

[27] C. Tranoris, K. Thramboulidis, "From Requirements to Function Block Diagrams: A new Approach for the Design of Industrial Applications", 10th Mediterranean Conference on Control and Automation, Lisbon. 9-12 July, 2002.

[28] Vyatkin, V.; Hanisch, H.-M.; Application of visual specifications for verification of distributed controllers, IEEE International Conference on Systems, Man, and Cybernetics, 2001 Volume 1, 7-10 Oct. 2001 Page(s):646 – 651

[29] ICS Triplex, Web Site, http://www.isagraf.com, 2006.