IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review

Valeriy Vyatkin, *Senior Member*, *IEEE*

Abstract — **This review paper discusses the industrial and research activities around the IEC 61499 architecture for distributed automation systems. IEC 61499 has been developed to enable intelligent automation where the intelligence is genuinely decentralized and embedded into software components, which can be freely distributed across networked devices. With the recent emergence of professionally made software tools and dozens of hardware platforms, IEC 61499 is getting recognition in industry. The paper reviews research results related to the design of distributed automation systems with IEC 61499, the supporting tools and the aspects related to the execution of IEC 61499 on embedded devices. The promising application areas of IEC 61499 include flexible material handling systems, in particular airport baggage handling, flexible reconfigurable manufacturing automation, intelligent power distribution networks and SmartGrid, as well as the wide range of embedded networked systems.**

Index Terms **-- IEC 61499, distributed automation, distributed embedded networking systems, intelligent automation systems**

I. INTRODUCTION

T he term "distributed automation" has been subsequently applied to three generations of technology. First, with the emergence of field area networks, it has become possible to collect data from sensors distributed across geographically dispersed areas, while their processing was done centrally in Programmable Logic Controllers (PLCs). Second, there were attempts to facilitate integration of PLCs into systems communicating via networks, by proposing integration component architectures, such as Modbus-IDA [1] and PROFInet-CBA [2]. Finally, it comes to the genuinely distributed automation development, where the intelligence is designed from the very beginning as decentralized and embedded into software components, which can be freely distributed across networked hardware devices. The design of distributed systems, in general, has been identified as a grand challenge of computing [3]. The IEC 61499 architecture [4] has been conceived to facilitate the use of distributed automation intelligence, but for some time the standard could not make its way to the industrial practice. Now, with the emergence of professionally made software tools and dozens of hardware platforms one can expect stronger industrial interest to the distributed automation.

First industrial applications of commercial IEC 61499 compliant tools and platforms confirm its benefits in terms of much improved design performance – i.e. the time and effort needed to develop and commission automation software. One such example is the use of ISaGRAF IEC 61499 implementation to automate the experimental shoe manufacturing factory [5, 6] shown in Figure 1. The factory was developed to achieve the revolutionary high level of manufacturing flexibility, thanks to which individually tailored shoes could be produced at cost of mass manufactured product.

Figure 1. Experimental shoe manufacturing facility at ITIA-CNR automated with IEC 61499.

New types of material handling devices called "terns" were developed for this purpose. The terns form easily configurable lines, called "molecular" for their shape. As seen from Figure 2(a), this new equipment allows to accommodate multiple product paths through the system, where each path corresponds to a particular sequence of operations. The IEC 61499 control program of the molecular line, presented in Figure 2(b), exhibits the high degree of code modularity, and therefore enables reusability and re-configurability of the equipment. As seen from Figure 2(b), the line controller uses six instances of the *TernControl* function block, each of which is a composite of three controller function blocks, representing constituent parts of the tern: *Table*, *Manipulator* and *Island*. Each of these, in turn, is another composite function block. At the lowest level in the hierarchy, basic function blocks are programmed using the state machines language (called Execution Control Charts, or ECC) and traditional PLC languages, such as Ladder Logic Diagrams.

The function blocks architecture of IEC 61499 has been formally voted and published in 2005. Since then, many useful practices have been earned by researchers worldwide. According to some researchers, many expectations seem to have failed [7] while the others see the situation in a more positive way [8].
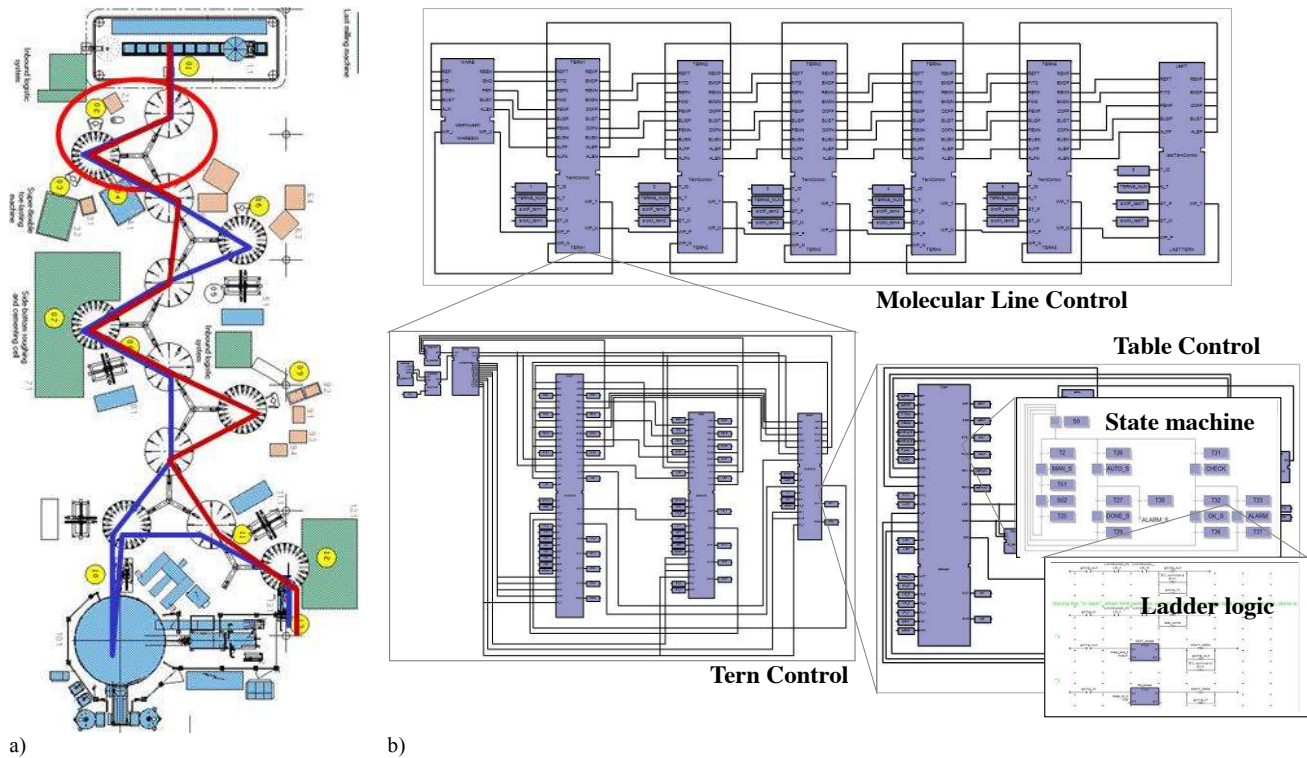
V. Vyatkin is with the Department of Electrical and Computer Engineering, University of Auckland, Auckland 1142, New Zealand (e-mail: v.vyatkin@auckland.ac.nz).

Figure 2. a) Product flows through the "molecular line", and b) function block control enabling the flexibility of the "molecular line".

Taking IEC 61499 as a reference architecture of future industrial automation information technology (IT), researchers have addressed virtually all sides of the IT scope and life cycle: from theoretical formal models to the business use case scenarios. The corresponding research publications can be roughly sorted into three following categories:

First of all, the research works focused on various aspects of the IEC 61499 implementation. The standard proposes an abstract executable model of distributed systems, leaving unanswered many questions related to the methods of its implementation. These issues needed to be addressed before the compliant systems could be developed, or were addressed during such development efforts. In this survey 20 works can be classified as belonging to this group.

Another large set of research works tried to prove the claimed benefits of IEC 61499 as compared to the solutions currently used in the field, for example, PLC architecture of IEC 61131-3 standard [9], or the general purpose languages implemented on industrial personal computers (IPCs) and embedded devices. This class also includes works on migration from other architectures to IEC 61499 and accounts for 40 works in this survey.

Many research works have been trying to introduce various trendy ideas from computer science into industrial automation context using IEC 61499 as a vehicle. To mention a few, one can cite object-oriented architectures and design patterns [10], in particular UML-based engineering [11-15], new execution models inspired by different formalisms [16, 17], formal verification, semantic Web technologies [18], web services, ontologies [19], and automatic program transformations (refactoring) [20]. In most cases, these research works demonstrate the potential of IEC 61499 to implement those techniques in a much more consistent way than any existing IT technology used in industrial automation. About 40 related works are cited in this survey.

It is worth mentioning the difference in attitudes to the standard by academic and industrial researchers and practitioners. Most practitioners take the standard as a serious normative document not allowing for any flexibility in interpretation of its provisions. On the other hand, some researchers (from the more theoretical end of the spectrum), easily propose fundamental changes to the nature of the standard, using it rather as the inspiration source for further enhancement in automation technology. This often creates confusion and difficulty in understanding the applicability of research results to industrial problems. This paper attempts to help by reviewing the state of the art in the IEC 61499 related research and its impact on the implementation of compliant tools, devices and applications. It follows the line of several surveys published recently e.g. [21-23], extending them with the fresh research results and discussing the older references from that particular perspective. There are some introductory books on the topic by Lewis [24], Lastra et al. [25] and the author [26]. Given the availability of introductory material on IEC 61499, in this paper no comprehensive introduction to this architecture is provided.

The rest of the paper is structured as follows. In Section II the key features of IEC 61499 are presented. Several concerns on IEC 61499 expressed by researchers and practitioners are discussed in Section III. One such concern related to semantic ambiguities of the IEC 61499 execution is discussed in more detail in Section IV. In Section V, the design related developments are presented. Section VI reviews the progress related to execution of function block applications. Section VII discusses verification and validation related research and development efforts. In Section VIII first industrial installations are mentioned and the motivation of early adopters is discussed. The paper is concluded with summary, outlook and the list of references.

II. FEATURES OF IEC 61499

A. Function block: process or code module?

The IEC 61499 architecture exploits the popular among control engineers block-diagram way of thinking. The main design artefact, function block (FB), has been extended from the subroutine-like structure in IEC 61131-3, to the process–like abstraction used in the theory of distributed computing systems. A process represents an independent computational activity with its own set of variables (context) and communication with other processes via messages. The event interface is well suited to modelling of inter-process message-based communication.

On the other hand, a function block still may represent just a piece of code executed within another process. However, the process-like encapsulation mechanism provides the strength to this architecture, enabling arbitrary re-allocations of components to distributed execution domains without affecting their functionality.

This duality of the function block language construct often creates confusion amongst researchers and practitioners. However, one should note that the standard provides a certain redundancy of structures and there are many ways to model same application configuration using IEC 61499 artefacts.

B. Model or implementation?

As one could see from the example in Figure 2, the architecture supports unlimited nesting of composite function block structures, and combination of several diagram types: block-diagrams, state charts, and ladder logic in the same design. The result of the design is an executable specification of distributed automation system, which includes also models of devices and their network interconnections.

The downside of this duality of being both model and executable implementation shows itself when it comes to portability: a particular executable FB configuration needs to include platform dependent service interface function blocks, which hinders the portability.

C. Event-driven execution

Function blocks of IEC 61499 are event-driven, i.e. they remain idle unless an event is sent to one of their event inputs. The main motivation for event-driven execution is portability, i.e. the desire to make the code independent of the sequence of FB invocation in the PLC scan loop. The event-driven execution is the key mechanism enabling transparent modelling of distributed systems.

After a FB is activated by an input event, it is assumed that it cannot be re-entered before the previous activation has terminated.

The event mechanism is a nice abstraction, but the problem is that it has to be applied uniformly. The standard does not answer questions related to what to do if events arrive too fast one after another. Theoretically, one can restrict the minimum time interval between the events originated in the process, but it can be different when the source of events is purely computational, originated in another function block executed in the same device. In this case the runtime environment has to implement either storage for events, or explicitly loose some.

D. Encapsulation

Another provision for portability is strong data encapsulation into components which has been widely recognized in the software community as one of the pillars of creating safe and re-usable code. It can ensure the absence of hidden dependencies between variables of several FBs. This model also reflects the fundamental property of distributed systems where any data exchange can be implemented only via explicit message passing.

As indicated in [27], this concept may seem inconvenient to many PLC developers who got used to work with shared variables. However, smarter design tools can make the development process easier. Education effort can be directed towards design patterns and examples showing how the typical automation problems are solved using function blocks without global variables.

III. CONCERNS

With the increasing complexity of information and control systems, the developers realize the need in distributed design languages and architectures. In this regard, the IEC 61499 architecture would be their first choice. However, there are some barriers preventing the practitioners from using IEC 61499, the main of which are the lack of: mature engineering tools, reliable embedded control hardware, proven design methodologies and trained engineers.

Reference industrial projects could be useful, but, again, with the increasing requirements to safety and security, it is hard to expect that such projects can be implemented with a seemingly superior, but still unproven technology like IEC 61499. In the rest of this section we discuss several concerns expressed by research and industrial community.

A. Determinism

Implementation of the event-driven activation of function blocks implies the possible need of storing events in queues of a variable length and loss of events in case of the queue capacity is exceeded. Altogether, this may lead to non-deterministic behaviour of the control device, i.e. different computation results in exactly same input conditions (for example, if timing of input signals in one run is a bit different from another run).

Attempts to increase the determinism of FB execution include propositions of the synchronous model of execution [28], cyclic model of execution [29, 30], and ISaGRAF model [31], which is close to the cyclic. These models "bend" a bit the fundamental concepts of event-driven invocation, implemented in the pure event-driven implementations, such as FBRT [32] and FORTE [33].

A possible application-level practical solution to achieve the determinism of event-driven applications is to sample external inputs periodically, making them available to the rest of the application if any change is detected (presented in Figure 3). Here one can see a pulse generator B1 ("E_CYCLE") which emits events periodically (each DT=10ms in this case). The events activate the FB B2 which samples the inputs from the corresponding hardware modules interfacing the process. In case of any input value has changed its value since the previous invocation, this FB emits the CHG event and activates the B3 function block (CONTROLLER). The latter recalculates the values of outputs and passes them to B4 which interfaces the hardware output module.
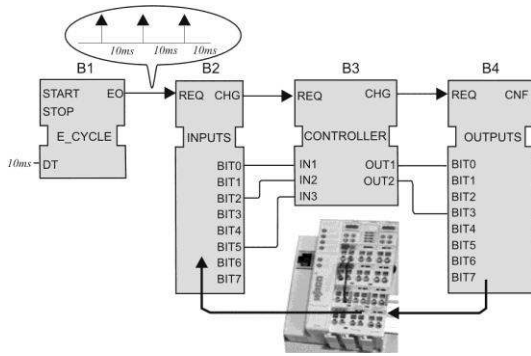
Figure 3. Function block application structure with periodic sampling of inputs.

While this solution does not seem to be as exciting as compared to the pure interrupt-driven input architecture (in which case the E_CYCLE FB would not be required), in fact it preserves the event-driven nature of the application (the block B3) and all associated benefits. However, the behaviour in this case is fully deterministic, provided the sampling period DT is sufficient to complete the execution of B2, B3 and B4. The choice of the sampling period shall be based on the "mechatronic capability" of the device [31], that can be determined empirically or using Nyquist – Shannon sampling theorem. On the other hand, to ensure determinism, the worst case execution time of the application (FB B3 "Controller" in this example) needs to be predicted in advance. The minimum value of the inputs sampling period should not be less than the worst case execution time of the application. This can be done by software tools through static timing analysis as proposed in [34].

This solution is different from the PLC scan as the controller application executes only when a change of any input is detected. The event-driven execution also can help in activating only those blocks which are directly dependent on that event, unlike the PLC case where all program modules are executed. This may require less time for execution even in the worst case.

Another benefit of this approach is that service interface function blocks for input modules can be developed using the usual polling of their values that is easily supported on most architectures, rather than requiring special hardware with interrupt-driven input update.

B. Best case, worst case, average case

It has been argued in [35] that there is no need in industrial automation to use the benefit of event-driven execution to improve the average or best case reaction time of the control device, as only the worst case time does really matter. Although this statement is true in general, there are some special cases.

First, there are many automation applications without hard-real time constraints, for example, building management systems, where the control device accommodates control of many processes. Improvement of the average reaction time in such systems will mean better quality of control and more cost-effective solution. In such systems a control reaction may involve communication between several nodes which has unpredictably variable duration anyways. Thus, in such networking environment, improvement of the local processing time of each node will favourably impact on the overall system performance.

One more reason for improvement of the best and average execution time is power conservation. Many battery powered control devices will exhibit much longer battery life in case of event-driven activation that leads to shorter and rarer CPU activity periods.

C. Performance

In a traditional PLC, performance is measured in terms of reaction time, which is bounded by the doubled scan time. The reaction time of the IEC 61499 compliant controller is harder to estimate, as it can vary significantly dependent on the source of input.

Many complaints on the low performance of IEC 61499 originate in incorrect association with the Java technology, which was used in the first IEC 61499 implementations such as FBRT environment [32]. In fact, there are many other (more efficient) implementation techniques as it will be discussed later in Sections III.F, VI.

Another group of performance-related concerns originates in the wide usage of eXtensible Markup Language (XML) in IEC 61499. XML is used in IEC 61499 as a storage format for all design artefacts, and even for implementation of device management protocol. The use of XML provides numerous benefits, as the representation becomes self-explanatory, and standard XML parsing tools can be used to check syntax. On the other hand, XML tools are quite performance hungry and therefore not appropriate for many embedded platforms. One solution was presented by Zoitl et al. in [36] who suggested using binary XML. This solution, however, impacts on interoperability, as there are several different versions of binary XML, supported by different user groups.

The precise implementation of the event mechanism of IEC 61499 also adds substantial computational overheads. To ensure the correct causality of execution, all events emitted in a function block application need to be sequentially ordered. A "first in, first out" (FIFO) queue has been proposed in several implementations, e.g. [37] to store all events emitted by component function blocks and dispatch them one by one from the top of the queue as shown in Figure 4.
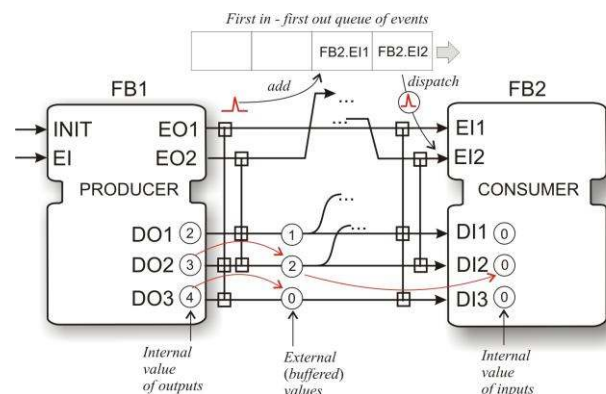

Figure 4. Implementation of event passing mechanism using FIFO queue.

It has been shown in [38] that data buffering is required to handle correctly the situations when event flow does not coincide with the data flow. Therefore, a buffer needs to be associated with each data connection. On an output event (such as FB1.EO2), all associated data are written to the buffers (for example FB1.DO2 value 3 is written to the buffer where it overwrites the previous value 2). On an input event (e.g. FB2.EI2), all associated input data values are read from the buffers into function blocks (including

FB2.DI2 which now receives the value 3). However, the queuing of events implies the possibility to loose events in case of the queue overflow. One should also note that the data buffers will be overwritten with the freshest value, so it may happen that an event will activate FB with data completely different from those which were calculated at the moment when the event was emitted.

The issue of guaranteeing hard-real time constraints of IEC 61499 applications has been comprehensively addressed by Zoitl in [37]. Kuo et al. [34] use formal modelling to estimate the longest computational path to predict worst case execution time.

D. IP Protection vs. Openness

Many practitioners raise the issue of intellectual property protection in IEC 61499 due to the open XML representation format.

Solutions of this problem are not different from other programming frameworks. Instead of supplying full source code of a function block whose content needs to be protected, the vendor can provide a library of blocks compiled for a particular platform. This can be supplied along with "abridged" XML representation containing only interface and, possibly, the service sequences describing the overall behaviour of the FB without revealing sensitive implementation details. Using the "compilation to hardware" technique, discussed later in Section VI, D, one can achieve even more protection of intellectual property.

An illustration of the service sequences mechanism of IEC 61499 is provided in Figure 5. The function block IN_EVENT (whose source code is hidden) implements a "button" on screen which can be used to interact with FB applications. The FB implements four operation sequences: normal establishment, normal termination, normal operation, and operation inhibited. The illustrated "normal operation" sequence describes the scenario which starts with a click or key pressing on the button. After some short delay, the output qualifier is set to 1 and event at the output IND is emitted.
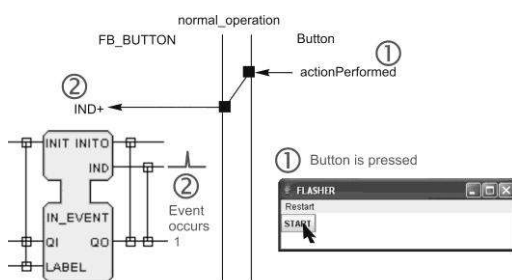

Figure 5. Description of function block functionality using service sequences [26].

Given the service sequences, one can partially reconstruct the execution logic of the function block and of the entire system without knowing sensitive details of the implementation.

E. Expressive power of design

The use of communicating state machines (ECCs of basic FBs) for programming distributed systems has been quickly identified by many researches and practitioners as a very convenient feature of IEC 61499.

However, one necessary part of programming badly supported in IEC 61499 is exceptions handling, that is the reaction of a program to some erroneous situations that can happen in any state. Defining these leads to the ECC cluttered with many connections to the "Exception" state and back (for the recovery).

The standard model of dealing with this problem is exemplified in Harel's State Charts [39], where hierarchical states are allowed. One can combine all states of the control algorithm to such a composite state and describe a single "exception transition" for the whole group.

Shaw et al. [40] proposed the use of hierarchical state machines in place of ECC in basic function blocks. This representation, called HCECC, however, requires non-compliant syntax thus affecting the portability. HCECC can be compiled to the standard "flat" ECC, but reverse engineering is not possible. In general, this solution seems to be overkill for solving the exception handling problem only. There is no doubt in the usefulness of hierarchical state machines as design artefact, but, in the IEC 61499 context similar effect is achieved using composite function blocks. Future software tools could support "hierarchical state machine" front-end with dual conversion to/from nested composite FB types.

F. The "chicken and egg" problem of tools

The standard has inspired many researchers to create the supporting software tools. The usual implementation tool set includes a *workbench* for editing function block designs and translating them into executable form, and some kind of *run-time environment*, that supports the execution of the executable code.

The most developed examples of such research-oriented workbenches are FBDK [32] and 4DIAC-IDE [33]. These have been supported with a consistent development effort until now, with 4DIAC-IDE being an open source project. Substantial development effort has been invested to CORFU/Archimedes [41] and another open source project FBench [42], but at the moment these tool do not seem to be continuously supported. There are a few others, less developed ones, like FBLab [43].

Runtime environments include FBRT [32], FORTE [33], FUBER [44] and Cyclic RT [29]. Some of the solutions, such as synchronous compiler [28], claim not to require any run-time support as they generate the complete executable file to be placed to the desired embedded target platform.

These tools have been successfully applied in many automation projects but mostly in academic and research labs. However, there is a barrier of using them in industry. The commercial PLC tools provide high level of design and remote debugging support that is hard to compete with. Achieving this level of maturity, acceptable for industry, requires years of development and improvements.

First attempts to overcome this barrier are commercial developments ISaGRAF [45] and NxtStudio [46].

ISaGRAF (since 2008, a Rockwell Automation company) combines the support of both standards IEC 61131-3 and IEC 61499, so it is possible to develop distributed control applications together with the use of PLC languages. This tool supports a very seamless way of code distribution to the networking devices. The workbench automatically inserts communication code where it is required, while the user sees only the global picture of the entire distributed application.

NxtStudio (developed by an Austrian company nxtControl) integrates distributed control approach based on

IEC 61499 with SCADA[1]. It is an industrial grade engineering environment which supports the design of control applications and visualization together in one tool. This approach has great advantages in productivity and reuse of both control and visualization components. Several features of NxtStudio have long been expected from IEC 61499, for example, the debugging and online-monitoring infrastructure, allowing to remotely debug single FBs as well as fully distributed applications. Another feature is the automatic generation of the communication during the distribution process of the application. This greatly reduces the engineering effort when distributed control applications are designed.

With these two powerful tools already on the market, the developers have sufficient choice in trying out the benefits of IEC 61499.

G. High educational threshold

Many practitioners afraid of that IEC 61499 requires a steep learning curve. This is partially true, as it introduces a lot of new concepts as compared to PLC technology. On the other hand, it is the author's personal experience that IEC 61499 is very easy to learn and use by the current generation of students. When young control engineers learn IEC 61499, many of them get a feeling of some déjà vu of something very intuitive and familiar they have been looking for but could not find among the available PLC programming technologies.

IV. SEMANTICS IMPROVEMENT EFFORT

A. Improvement of the standard

Some semantic ambiguities of the standard were spotted for the first time during the period of standard's industrial and academic evaluation. In [47] it was illustrated that definition of input event variable lifetime in basic function block can result in different behaviour of control system. In [48] different scheduling policies of function blocks in composite structures were demonstrated.

The standard was voted in 2005, and its final text was modified based on some of the findings. For example, an early draft of the standard was prescribing to store a copy of event input variable value (so called *latch*) in a basic function block implementation. Because it was shown in [47] that the latches still can lead to a loss of input events, they were completely removed from the text.

In subsequent works, e.g. [44, 49-51], the ambiguities of the IEC 61499 semantics were classified and analysed in detail, showing possible impact of different interpretations on the correctness of control applications. Strasser et al. [52] and Doukas et al. [53] investigate the semantics of device, resource and of a distributed system. In more detail the semantics improvement is covered in [54].

As a result of this semantics improvement effort, the second edition of the standard voted in 2010 included a number of corrections which eliminated the ambiguities.

B. Execution differences

However, the original semantic ambiguities of the standard gave rise to a number of execution models such as the cyclic and synchronous execution models which attempt to "combine" the event-driven execution of IEC 61499 with

determinism of the cyclic scan semantics of PLCs. These have been implemented in several run-time environments, including ISaGRAF. However in such models the following problematic situations can occur:
- Several events can arrive to the inputs of the same FB simultaneously;
- The syntax of IEC 61499 does not allow to detect the simultaneity of events inside the FB;

As a result, a FB, written for such models may run differently in other execution environments and vice versa. We will illustrate the impact of semantic differences on the following simple example of a baggage handling system (BHS).

Example. As illustrated in Figure 6, two conveyors C1 and C2 merge bag flows to conveyor C3. The bag tracking function of this BHS is implemented in a distributed way, so that one function block is responsible for knowing the bags' location within its respective conveyor section as shown in Figure 7,a (more on distributed BHS control in [55] and [56]). When a bag leaves the conveyor to the downstream conveyor, its record is passed to the corresponding downstream function block.
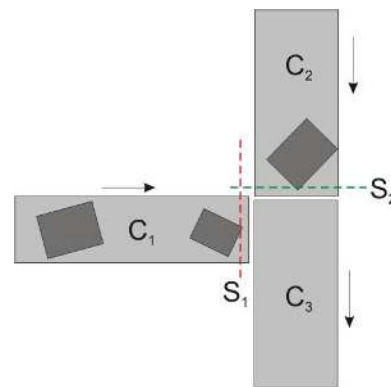

Figure 6. Sample merging conveyor structure.

A bag record is implemented using a user defined structured data type. For example, to model the bag passage from C1 to C3, the C1 FB will assign the bagRO variable to the value of the bag record leaving the conveyor. On event of S1 sensor goes high, C1 will emit event EO, connected to the event input EI1 of C3. Having received the event at its EI1 input, the FB C3 will get activated and will add the bag record to its internal data base. Similarly the bag is passed from C2 to C3. The internal logic of the mrg_conv function block implements the processing of the arriving bags in the state machine shown in Figure 7, b. Here the algorithms ADD1 and ADD2 add to the internal database of this conveyor the ID of the bag arrived from C1 or C2 respectively.
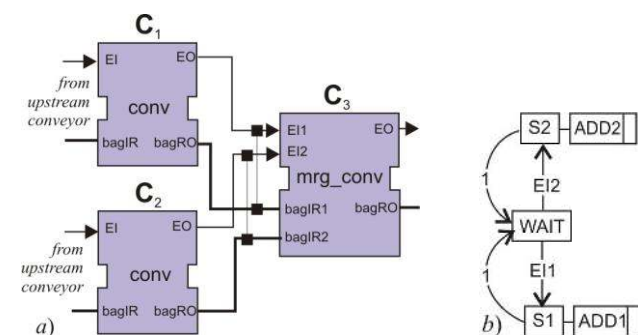

Figure 7. Distributed control of the merging conveyors in IEC 61499 function blocks.

In the "pure" event-driven implementation platforms, such as FORTE, NxtForte and FBRT, in a rare, but possible case of simultaneous detection of bags by sensors S1 and S2, two events will appear at the input interface of C3. The function block C3 will be invoked twice, and, as a result, no bag record will be lost.

In the cyclic or synchronous semantics both events arrive simultaneously and the C3 block gets activated only once. The priority of input event processing will determine which record will be added to the database and which one will be potentially discarded and lost.

To be able to resolve this situation within `mrg_conv` FB, it is required to detect the situation of simultaneous arrival of events to the FB inputs. It could be done as shown in Figure 8. Here the transition WAIT -> S12 has condition that includes both input events: EI1 and EI2. However, the IEC 61499 syntax fundamentally prohibits the use of two event names in the condition transition.
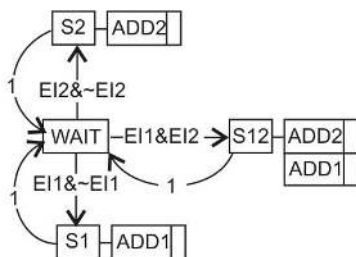

Figure 8. "Impossible ECC" which could detect the simultaneity of input events.

V. DESIGN

The natural expectation from the system-level architecture like IEC 61499 is that it improves considerably the designer performance, which is expected to result from increasing the level of design, i.e. using design artefacts of higher abstraction level.

A. Redefining model-driven design

Model-driven engineering is the major trend in software engineering. For example, UML technologies and tools are widely used in all sectors of industry to support the design of complex software systems. Matlab/Simulink is used in embedded systems design for the same purpose.

The function blocks architecture of IEC 61499 is clearly following these trends, providing the developer a mix of models that include distributed system model composed of devices, device model composed of independent resources, application model as a block diagram composed of function block instances, basic function block model specified as a state machine, and algorithm model that can be specified in any programming language, including those of PLC world.

An early survey of related modelling techniques can be found in [57]. Bonfé and Fantuzzi [58] and Thramboulidis [12] have introduced the use of UML in automation and in particular in the IEC 61499 context. The latter work proposed generation of function blocks from UML diagrams, while Dubinin et al. proposed the UML-FB architecture with both ways of generation of UML diagrams from function block designs and vice versa in [59]. In [60] Thramboulidis proposed IEC 61499-based concept of model-integrated mechatronic architecture for automation systems design. Panjaitan in [61] demonstrated the use of UML for addressing a number of practical issues related to automation systems design with IEC 61499. An interesting comparison of the IEC 61499 modelling artefacts with statecharts and Petri nets was done by Barji et al. in [62].

SysML is a UML derivative for engineering applications that is getting increasingly popular. In particular, SysML is well supporting such design phases as requirements capturing and formalization of specifications. Hirsch et al. [63, 64] provide a pathway for linking function block technology with SysML.

One can summarize the rationale of combining UML/SysML and IEC 61499 as shown in Figure 9. The graph represents developer performance through system engineering steps with both types of technologies. UML/SysML better support the top-down design and requirements engineering. However, they are less efficient when it comes to the deployment to distributed embedded targets. Also legacy PLC programming (such as ladder logic) is not supported there. As illustrated in Figure 9, the efficiency of SysML development is high in the early design stages and lower in the deployment and maintenance.

On the other hand, the IEC 61499 distributed architecture, copes well with the problems of code generation, deployment and reconfiguration, but offers less support at the initial design steps. Their combination intends to bring best of both worlds as it is conceptually illustrated in Figure 9. One should note that it is very hard to quantify the impact of software methodologies.
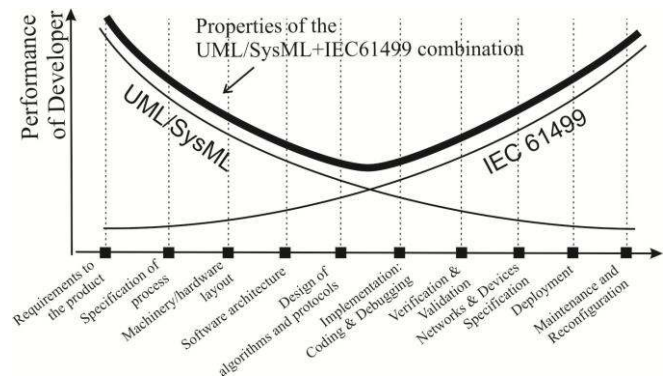

Figure 9. Desired properties of the UML/SysML + IEC 61499 combination.

However, there is one more reason to call the design in IEC 61499 framework *model-driven*. Christensen has proposed in [10] a number of design patterns, among which the adaptation of Model-View-Control to IEC 61499 and industrial automation. In this context the Model is understood as a precise behavioural model of the plant. An example of MVC use can be found in [65], Ch. 13-16. The concept of Automation Object (AO) [18, 66, 67] is an attempt to generalize the FB concept to represent a machine or a mechatronic part thereof. The Intelligent Mechatronic Component (IMC) [68] concept is an AO implemented using the MVC pattern, thus enabling simulation and visualization along with deployment of controls.

B. NxtControl concept of CAT

The MVC design pattern has motivated NxtControl to invent the *composite automation type* (CAT) concept.

As illustrated in Figure 10, a CAT is a function block that combines functions of machines or their parts, with their simulation and visualization. Here the "Pick and place" manipulator is built of two identical pneumatic cylinders, each represented by a CAT in the FB application.
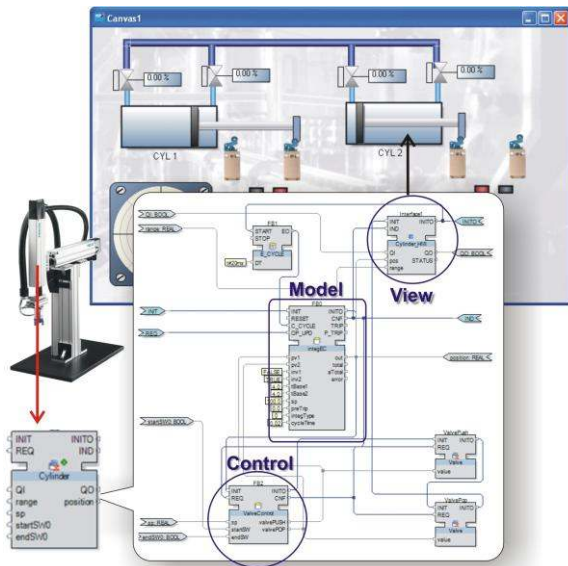
Figure 10. The concept of composite automation type in NxtStudio is based on the MVC architecture.

Once the application is assembled from instances of such CATs, NxtStudio can automatically deploy the control parts of all CATs to the designated embedded targets, while the View parts will be sent to the device displaying SCADA screens. In the figure, a CAT of a pneumatic cylinder is exemplified. The CAT also includes the (behaviour) model of cylinder's dynamics. Once executed, the application built of these CATs immediately delivers complete interactive simulation model of the manipulator.

The CAT concept has proven its benefits in a number of industrial projects, where NxtControl tools were used, for example in building management systems automation.

C. Migration from PLCs to IEC 61499

The huge legacy of industrial automation raises the issue of software *migration* from the existing PLC platforms to the IEC 61499. The practitioners, for example, Woll [69], are concerned with the prospect of complete paradigm change in automation software design and want to secure their investments in automation hardware and software. Therefore, investing into the migration pathways becomes crucial for the industrial uptake of IEC 61499.

The migration case studies were presented by Hussain [70] and Gerber [71] with some recommendations for manual migration provided. Wenger [72] presents an attempt of converting the entire PLC application to an equivalent IEC 61499 one. The result of this approach may run correctly, but would not be very usable in the entire project's life cycle due to poor structure and readability of the generated code.

Dai in [73] presents an example of domain-specific migration related to baggage handling systems (BHS). Here the emphasis is made on generating well designed modular code which can be directly identified with the equipment it controls. Shaw [74] also investigates migration of the PLC based BHS automation software to IEC 61499 with subsequent compilation to C.

In addition to the migration research, the work [75] discusses harmonization of IEC 61499 and IEC 61131-3 – in terms of execution semantics and data types, aiming at seamless use of PLC languages inside IEC 61499 function blocks.

D. Integration with domain specific standards

The strong benefit of the system level design language like IEC 61499 is its capability to implement the provisions of domain specific design practices standardized in a plethora of international standards. According to [76], the open nature of IEC 61499 can help creating open knowledge economy in industrial automation. There are numerous examples of ongoing research works proving this assumption.

In the process automation domain, Peltola et al. [77] and Dimitrova et al. in [78] explored the concept of integration with ISA88 standard. Lepuschitz and Zoitl [79] consider integration with the IEC 61512 standard, providing models and terminology for batch control.

Pang [80] explored application of CAEX and IEC 62424 for using physical layout of automation systems for automatic code generation.

Higgins et al. in [81] proposed a way to combine IEC 61850 standard from power distribution automation domain with function blocks of IEC 61499. As demonstrated in subsequent works, this solution can help achieving many of SmartGrid functions.

Hagge and Wagner [82] studied implementation of OMAC state machines defining standard controller modes in manufacturing applications.

Colla et al. [83] report on domain-specific modelling in the shoe manufacturing sector.

E. Flexibility, reconfigurability and agent-based-design

Increasing of flexibility and re-configurability of industrial systems has been one of the major drivers of IEC 61499. Many motivations have been derived from the Holonic manufacturing systems international program [84] promoting multi-agent control.

It has been discovered by researchers that requirements to flexibility of manufacturing systems often can be implemented by reconfiguration of automation hardware and software. Correspondingly, several works address various aspects of such reconfiguration using IEC 61499 and addressing problems "under the hood" of the compliant programmable controllers. One of the first works in this direction was by Fletcher and Norrie [85], followed by a number of works from Technical University of Vienna, in particular addressing provisions of IEC 61499 architecture by Strasser et al. [86], real-time implementations by Zoitl et al. [87] and on down-timeless evolution by Rooker et al. [88].

Another group of works focuses on end-user aspects of manufacturing facility re-configuration with IEC 61499. These include, for example, works on agent-based reconfiguration by Brennan et al [89], intelligent ontology-based infrastructure supporting reconfiguration by Alsafi and Vyatkin [90], reconfiguration protocol by Khalgui and Hanisch [91], CNC reconfiguration by Huang [92] and even intelligent energy supply environment for flexible manufacturing by Budhia et al. [93].

A number of laboratory and pilot industrial installations explored the combination of IEC 61499 distributed architecture with multi-agent control. The fully distributed approach to baggage handling systems automation was developed by Black in [55]. Lepuschitz et al. [94] proposes a hierarchical multi-agent architecture based on IEC 61499 which enable elements of self-configuration in manufacturing systems, and Hegny [95] investigates the use

of IEC 61499 to implement multi-agent control in material handling systems. Huang [96] discusses the architectural solutions for joining IEC 61499 lower-level agents into upper multi agent manufacturing platform.

Despite the number of works in this direction, still there are many open questions on the link between IEC 61499 and the agent-based automation. In particular, the mechanisms of creating dynamic emergent structures in IEC 61499 need to be better addressed.

On the other hand, the model of IEC 61499 can be understood as a snapshot of distributed system software state at any moment of time. Evolutions between such states can be implemented on top of the IEC 61499 architecture, or even by its own means.

VI. EXECUTION

By definition, IEC 61499 is executable specification of distributed automation systems. Naturally, to enjoy the benefits of being directly executable (as opposed to more abstract design languages), one needs a tool chain which generates executable machine code from the IEC 61499 design artefacts. The tool chain needs to include the following component software tools:

- Compiler from the source FB format to an intermediate code executed with a virtual machine, or directly to machine code;
- Run-time environment – usually a set of libraries of function blocks implementing service functions akin to device drivers, responsible for scheduling of FB invocation, data and control flow and interfacing the peripherals;
- Support of device management protocol – the function implementing the load of FB application to a device, creation of new FB instances, or their modification.

The following hardware implementation platforms deserve mentioning:

A. On embedded controllers and PCs

The first attempts to execute IEC 61449 on commercially available hardware platforms date back to 2001-2002. Netmaster was a popular Java enabled controller used in several research labs, e.g. as reported in [97]. FBRT was used as software run-time environment.

Lastra et al. [30, 98] presented a run-time environment that executes IEC 61499 systems on the Java enabled controller of Nematron. The execution model can be classified as the cyclic, and the source code is compatible with FBDK format. The wide range of Nematron I/O modules is supported by libraries of service interface function blocks.

The 4DIAC-IDE workbench and FORTE runtime environment have been successfully used to deploy code to a number of embedded devices, e.g. Digi and PC/104 embedded controller [99].

Recently, execution of IEC 61499 has been demonstrated on a range of commercially available platforms, for example from Beckhoff, WAGO, SIEMENS and Advantech with commercial tools by ISaGRAF and NxtControl.

B. On standard PLCs

Ferrarini et al. [100] reported on the possibility to generate code for SIEMENS PLCs which can be further maintained with proprietary PLC tools. The approach is limited as it does not support the full set of IEC 61499 artefacts.

Shaw et al. [74] demonstrated that FB code can be executed on an Allen-Bradley controller programmable in ladder logic.

C. In specific proprietary environments

Polakow [101] presented LabView based execution of IEC 61499. LabView tool chain can take care of further code deployment to CompactRio embedded controllers. Obviously, such implementations can only partially fulfil the compliance requirements of IEC 61499, but allow to take advantage of the available and proven hardware and software platforms.

D. Direct hardware synthesis

Sullivan et al. [102] investigated implementation of IEC 61499 which leads to hardware synthesis in the form of a system on chip. The tool chain translates XML descriptions of function blocks and structures built thereof in VHDL, from which hardware is directly synthesizable. Possible benefits of this solution are as follows.

One can imagine the top-down design process for a distributed automation system, where for some devices a system on chip will be synthesized instead of deploying code to an existing microprocessor-based platform.

E. Performance and scalability

Chouinard et al. [35] report on implementation of a distributed system of 70 controllers demonstrating hard real-time operation when communicating over Ethernet.

Yan [56] presented a baggage handling system implementation with a few hundreds of composite function blocks deployed across 50 controller nodes communicating via Ethernet. Each function block was containing an intelligent controller of one conveyor section, which can achieve the BHS functions collaboratively, without any central supervisor. The measurements show less 0.1% load of the network used, which confirms the feasibility of fully distributed approach to control design.

F. Networking

A very important part of distributed systems design is related to network communications. By definition, IEC 61499 is compatible with any communication protocol, which is achieved by libraries of the corresponding communication interface function blocks (CIFB). There are two abstract communication patterns suggested in the standard already: CLIENT – SERVER and PUBLISH-SUBSCRIBE, which can be implemented by means of a particular protocol. For example, the first reference implementations were providing communication FBs based on the TCP/IP protocol stack.

In [103] Weehuizen et al. discuss the details of an implementation of the CIP protocol interface FBs. NxtControl has implemented various CIFB libraries to support communication over popular fieldbuses, such as EtherCAT and Profibus.

A comprehensive insight into the networking side of the IEC 61499 applications is presented in [104], in particular discussing CIP and MODBUS protocols implemented in IEC 61499 framework.

The work on integration of IEC 61499 with the "CAN in Automation" protocol has started in 2011 and is in progress [105].

VII. Verification and Validation

Verification and validation of automation software is a challenging task in general, but it is especially hard for distributed systems and event-driven function blocks.

The MVC architecture and its derivatives have been helpful in bringing together simulation and control design with IEC 61499. Yang [106] investigated the ways to use MATLAB/Simulink models together with controls written in IEC 61499. The use of such co-simulation environment has been reported in [107] for modelling distributed intelligent control of SmartGrid.

Formal methods of verification have been seen by many researchers as an efficient alternative to manual debugging and simulation based verification. The idea of formal verification is to prove rigorously (with the help of software tools) that certain properties hold in the execution of control system. The survey [108] presents the landscape of works on formal modelling and verification of IEC 61499 in the past decade.

New works include [109] by Čengić et al., who presented an attempt to model various execution models of IEC 61499. Khalgui et al. [91] use IEC 61499 as a platform for multi-agent control and formal verification. Allen [110] uses IEC 61499 to achieve flexibility of manufacturing control and explores its robustness by means of formal verification.

The work [68] develops the concept of Intelligent Mechatronic Components to enable systematic design of formal models used in the process of formal verification and paves the way to integration of simulation and formal verification.

One hard problem for formal verification is modelling data-rich computations. Pang [111] and Gerber [112] demonstrate the methods of extending the traditional discrete state formal languages with ability to model computations over data. This allows automatic model generation from arbitrary function block programs.

Suender et al. demonstrate in [113] the use of formal modelling and verification of IEC 61499 compliant devices to prove properties of reconfiguration, when a function block program is modified "on the fly" while it is performing control of the plant. That work is notable for proposing a framework for modelling not only a controller program but multiple layers of embedded device, including hardware, operating system, runtime and application. On top of that, reconfiguration scenarios are applied.

The recent progress of formal verification research gives hope to see the tools of commercial strength in the near future.

VIII. Proving benefits: Case studies and Industrial adoption

A. Industrial installations

The first industrial deployment of IEC 61499 compliant device was reported by Tait in [114] in 2005 at the meat processing plant at Rangitikei, New Zealand. The device was developed by TCS-NZ and based on ARM CPU, running Java based FBRT run-time environment. Since then, a number of pilot and commercial installations have been completed using IEC 61499 and the interest is growing.

NxtControl [46] reports on a number of commissioned building management systems where distributed control and visualization of the entire building was implemented using IEC 61499. The largest project has been a training center building with 19 control devices controlling about 2500 I/Os (heating, ventilation, air-condition, lighting, etc.) with IEC 61499.

Experience of system integrators has fully proven the design benefits of IEC 61499 compared to other automation technologies. In addition to the earlier mentioned shoe manufacturing line at ITIA-CNR, Colla et al. [115] investigated the application and implementation methodologies of IEC 61499 as enabler of flexibility in that sector.

There are close to one million of ISaGRAF compatible controllers deployed worldwide most of which have the IEC61499 capability (from version 5). Very few of the current ISaGRAF users are aware of this fact, though, which constitutes great opportunity to improve their design performance without extra investment.

Investigations on the use of IEC 61499 in the process control area are presented in [77, 79, 116].

There have been attempts to use IEC 61499 as a system level design and implementation architecture in the pure embedded control domains. For example, Insauralde [117] presented a case study in the area of avionics.

The vendors of the IEC 61499 technology are rapidly moving to various manufacturing sectors. Thus, NxtControl has demonstrated in 2010 the Delta robot prototype (Figure 11) fully controlled with the IEC 61499 technology using distributed control hardware of Siemens, Beckhoff and Wago, communication via Ethernet and EtherCAT and including servo motion control. This solution proves applicability of IEC 61499 for control of high speed manufacturing machinery.

The time-synchronisation technology, based on the IEEE 1588 standard can be very helpful in advancing distributed control to this sector and the first solutions have been demonstrated in [118].
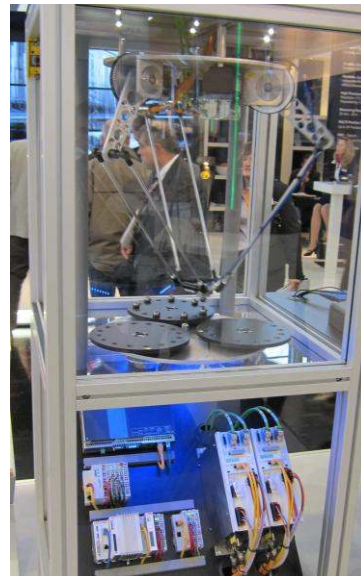

Figure 11. Delta robot demonstrating fitness of the NxtControl's implementation of IEC 61499 to control manufacturing machinery [119].

B. Motivations for adoption

The sectors where IEC 61499 has been applied include building management systems, process control, assembly, food processing and material handling. The main motivations for the IEC 61499 adoption currently spotted in industrial projects are as follows:

- Some system integrators have developed in house new hardware platforms for distributed automation and are looking for appropriate software. IEC 61499 is the only option if the applications are genuinely distributed, therefore, portable IEC 61131-3 solutions are not sufficient.
- Many system integrators have found standard PLCs overpriced as compared to industrial PC and embedded controllers. Therefore, the existing commercial tools combined with off-the shelf available hardware create a feasible alternative to the use of PLCs in many sectors of applications. However, there are sectors of automation, where the use of PLCs is de-facto standard, recommended to end users by the engineering consultancies. The situation is rapidly changing, however, with the arrival of new generation of industrial PCs or PACs (programmable Automation Controllers) which in many respects are as good as PLCs (e.g. in terms of supported I/O and communication interfaces and reliability), the same time having higher performance, lower costs and ability to be programmed with the new generation software technology.

C. Compliance: how strict it should be?

The IEC 61499 standard combines several technologies, targeting portability, configurability and interoperability of automation systems (see [26] for more detailed discussion). The first industrial implementations have already shown that industry is not ready to follow all these provisions simultaneously. Driven by market reasons, implementers select priorities in addressing different target features of the IEC 61499 technology. Thus, ISaGRAF has been more concerned in extending the system-level engineering capabilities of their tool chain, rather than achieving lower level code portability through the use of XML as program representation. Therefore, this tool chain has phenomenal capabilities in distributed deployment of large complex applications, but syntax and representation are proprietary and not compatible with other IEC 61499 tools at this stage. Compatibility with the legacy IEC 61131-3 run-time environment influenced ISaGRAF's decision to use the cyclic execution model, which differs from the execution model prescribed in the standard. However, one should note that these incompatibilities can be overcome in the future by developing software tools for automatic conversion.

On the other hand, NxtControl is more diligently following the syntax and representation prescriptions, but also had to extend these to implement a more powerful and user friendly visual editor and the CAT concept, combining control and visualization in one function block.

IEC 61499 includes the mechanism to handle and implement extensions, called *compliance profiles* [120]. By publishing such a profile, a vendor will explain the standard extensions introduced in a particular implementation. This mechanism helps in establishing self-management of the standard and related development activities. There are already a number of profiles published [121-123].

IX. CONCLUSIONS

The uptake of IEC 61499 has been rather long due to the fact this technology is new unlike the previous IEC 61131-3 standard which was the recognition of past practices. However, the extensive research effort in the past decade secured the very solid baggage of results to be taken and used by industry. Now first results can be observed in the market.

With the first commercial compliant tools and devices, users can already enjoy the numerous "design time benefits" of the IEC 61499 tool chains for system-level design and implementation of automation systems. Other benefits of IEC 61499, such as portability and interoperability have not been the major concern of the vendors yet. However, these features are "embedded" and hopefully will be recognized in a short time span.

The wider adoption of IEC61499 will help the industry to benefit from the promise of holonic and intelligent automation research results, providing the platform for deploying distributed applications with dozens, or even thousands of communicating autonomous control nodes.

The very promising development related to IEC 61499 is integration with domain specific design standards, such as IEC 61850 and IEC 62424, which may lead to integrated design methodologies where the control system can be automatically generated from the design documentation of other physical system parts.

Last but not least, the IEC 61499 standard has substantially contributed to the knowledge of distributed systems design in the industrial automation domain by providing the adequate notation and architecture that is complementary to the PLC programming architecture of IEC 61131-3.

X. BIOGRAPHY

Valeriy Vyatkin (SM'04) is Associate Professor and Director of the InfoMechatronics and Industrial Automation lab (MITRA) at the Department of Electrical and Computer Engineering, The University of Auckland, New Zealand. He graduated with the Engineer degree in applied mathematics in 1988 from Taganrog State University of Radio Engineering (TSURE), Taganrog, Russia. Later he received the Ph.D. (1992) and Dr Sci degree (1998) from the same university, and the Dr Eng. Degree from the Nagoya Institute of Technology, Nagoya, Japan, in 1999. His previous faculty positions were with Martin Luther University of Halle-Wittenberg in Germany (Senior researcher and lecturer, 1999–2004), and with TSURE (Senior Lecturer, Professor, 1991–2002).

His research interests are in the area of dependable distributed automation and industrial informatics, including software engineering for industrial automation systems, distributed architectures and multi-agent systems applied in various industry sectors: SmartGrid, material handling, building management systems, reconfigurable manufacturing, etc. Dr Vyatkin is also active in research on dependability provisions for industrial automation systems, such as methods of formal verification and validation, and theoretical algorithms for improving their performance.

XI. REFERENCES

[1] J. Camerini, A. Chauvet, and M. Brill, "Interface for distributed automation: IDA," in *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, 2001, pp. 515-518 vol.2.

[2] K. Trkaj, "Users introduce component based automation solutions," *Computing & Control Engineering Journal*, vol. 15, pp. 32-37, 2004.

[3] T. Hoare and Milner, "Grand challenges for computing research," *The Computer Journal*, vol. 48, p. 49, 2005.

[4] IEC61499-1, "Function Blocks - Part 1 Architecture," International Electrotechnical Commission, Geneva, International standard2005.

[5] A. Brusaferri, A. Ballarino, and E. Carpanzano, "Reconfigurable Knowledge-based Control Solutions for Responsive Manufacturing Systems," *Studies in Informatics and Control (SIC)*, vol. 20, pp. 31-42., 2011.

[6] E. Carpanzano and F. Jovane, "Advanced Automation Solutions for Future Adaptive Factories," *Annals of the CIRP*, pp. 435-438, 2007.

[7] K. Thramboulidis, "IEC 61499 Function Block Model: Facts and Fallacies," *IEEE Industrial Electronics Magazine*, vol. 3, pp. 7-23, Dec 2009.

[8] A. Zoitl and V. Vyatkin, "IEC 61499 Architecture for Distributed Automation: The "Glass Half Full" View," *IEEE Industrial Electronics Magazine*, vol. 3, pp. 7-22, Dec 2009.

[9] International Electrotechnical Commission, *Programmable Controller - Part 3: Programming Languages, IEC 61131-3 Standard*. Geneva: International Electrotechnical Commission, 1993.

[10] J. H. Christensen, "Design patterns for systems engineering with IEC 61499," in *Verteilte Automatisierung - Modelle und Methoden für Entwurf, Verifikation, Engineering und Instrumentierung*, Magdeburg, Germany, 2000.

[11] W. Zhang, W. A. Halang, and C. Diedrich, "Specification of Function Block Applications with UML," in *Robotics and Automation, Proceedings of the IEEE International Conference on*, 2005, pp. 4002-4007.

[12] K. C. Thramboulidis, "Using UML in control and automation: a model driven approach," in *Industrial Informatics, 2nd IEEE International Conference on*, 2004, pp. 587-593.

[13] S. Panjaitan and G. Frey, "Combination of UML Modeling and the IEC 61499 Function Block Concept for the Development of Distributed Automation Systems," in *11th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2006)*, 2006, pp. 766-773.

[14] T. Hussain and G. Frey, "UML-based Development Process for IEC 61499 with Automatic Test-case Generation," in *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*, 2006, pp. 1277-1284.

[15] V. Dubinin and V. Vyatkin, "UML-FB a Language for Modeling and Implementation of Industrial-Process Measurement and Control Systems on the Basis of IEC 61499 Standard," in *6th International Conference on New Information Technology and Systems (NITiS' 04)*, Penza, Russia, 2004, pp. 77-83.

[16] N. Hagge and B. Wagner, "Applying the handler-based execution model to IEC 61499 basic and composite function blocks," in *4th IEEE International Conference on Industrial Informatics (INDIN'06)*, Singapore, 2006, pp. 18-24.

[17] V. Vyatkin, "Execution Semantic of Function Blocks based on the Model of Net Condition/Event Systems," in *IEEE International Conference on Industrial Informatics*, 2006, pp. 874-879.

[18] O. J. Lopez Orozco and J. L. M. Lastra, "Adding function blocks of IEC 61499 semantic description to automation objects," *IEEE Conference on Emerging Technologies & Factory Automation*, pp. 809-816, 2006.

[19] Y. Al-Safi and V. Vyatkin, "An Ontology-based Reconfiguration Agent for Intelligent Mechatronic Systems," in *4th International Conference on Holonic and Multi-agent systems in Manufacturing*, Regensburg, Germany, 2007.

[20] V. Vyatkin and V. Dubinin, "Refactoring of Execution Control Charts in Basic Function Blocks of the IEC 61499 Standard," *IEEE Transactions on Industrial Informatics*, vol. 6, pp. 155-165, May 2010.

[21] K. H. Hall, R. J. Staron, and A. Zoitl, "Challenges to Industry Adoption of the IEC 61499 Standard on Event-based Function Blocks," in *Industrial Informatics, 2007 5th IEEE International Conference on*, 2007, pp. 823-828.

[22] K. Thramboulidis, "IEC 61499 in Factory Automation," in *IEEE International Conference on Industrial Electronics, Technology and Automation (CISSE-IETA'05)*, Bridgeport, USA, 2005.

[23] A. Zoitl, T. Strasser, K. Hall, R. Staron, C. Sunder, and B. Favre-Bulle, "The past, present, and future of IEC 61499," in *Holonic and Multi-Agent Systems for Manufacturing, Proceedings*. vol. 4659, ed, 2007, pp. 1-14.

[24] R. W. Lewis, *Modelling control systems using IEC 61499: Applying function blocks to distributed systems*: Iet, 2001.

[25] J. L. M. Lastra, A. Lobov, L. Godinho, and A. Nunes, *Function Blocks for Industrial-Process Measurement and Control Systems: IEC-61499 Introduction and Run-time Platforms*: Tampere University of Technology, 2004.

[26] V. Vyatkin, *IEC 61499 function blocks for embedded and distributed control systems design*: Instrumentation Society of America, 2007.

[27] W. Dai and V. Vyatkin, "Efficient Data Handling Model in IEC 61499 Function Block Approach," presented at the 8th International IEEE Conference on Industrial Informatics, (INDIN'10), Osaka, Japan, 2010.

[28] L. H. Yoong, P. S. Roop, V. Vyatkin, and Z. Salcic, "A Synchronous Approach for IEC 61499 Function Block Implementation," *IEEE Transactions on Computers*, pp. 1599-1614, 2009.

[29] P. Tata and V. Vyatkin, "Proposing a novel IEC61499 Runtime Framework implementing the Cyclic Execution Semantics," in *7th IEEE International Conference on Industrial Informatics*, 2009, pp. 416-421.

[30] J. L. M. Lastra, A. Lobov, and L. Godinho, "Closed loop control using an IEC 61499 application generator for scan-based controllers," in *10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05)*, 2005, pp. 323-330.

[31] V. Vyatkin and J. Chouinard, "On Comparisons of the ISaGRAF implementation of IEC 61499 with FBDK and other implementations," in *6th IEEE International Conference on Industrial Informatics*, 2008, pp. 264-269.

[32] Holobloc Inc. (2008). *Function Block Development Kit (FBDK)*. Available: http://www.holobloc.org/

[33] 4DIAC. (2010). *Framework for Distributed Industrial Automation (4DIAC)* Available: http://www.fordiac.org

[34] M. Kuo, L. H. Yoong, S. Andalam, and P. S. Roop, "Determining the worst-case reaction time of IEC 61499 function blocks," in *8th IEEE International Conference on Industrial Informatics (INDIN'10)*, 2010, pp. 1104-1109.

[35] J. Chouinard, J. Lavallée, J.-F. Laliberté, N. Landreaud, K. Thramboulidis, P. Bettez-Poirier, F. Desy, F. Darveau, N. Gendron, and C.-D. Trang, "An IEC 61499 configuration with 70 controllers; challenges, benefits and a discussion on technical decisions," in *IEEE Conference on Emerging Technologies and Factory Automation*, Patras, Greece, 2007.

[36] A. Zoitl, I. Hegny, and A. Schimmel, "Utilizing Binary XML Representations for Improving the Performance of the IEC 61499 Configuration Interface," *7th IEEE International Conference on Industrial Informatics (INDIN'09)*, pp. 66-71, 2009.

[37] A. Zoitl, *Real-Time Execution for IEC 61499*: ISA, 2009.

[38] V. Dubinin and V. Vyatkin, "Towards a Formal Semantic Model of IEC 61499 Function Blocks," in *IEEE International Conference on Industrial Informatics*, 2006, pp. 6-11.

[39] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, pp. 231-274, 1987.

[40] G. D. Shaw, P. S. Roop, and Z. Salcic, "A hierarchical and concurrent approach for IEC 61499 function blocks," in *Emerging Technologies & Factory Automation, IEEE Conference on*, 2009, pp. 1-8.

[41] K. Thramboulidis, "Development of Distributed Industrial Control Applications: The CORFU Framework," in *4th IEEE International Workshop on Factory Communication Systems*, Sweden, 2002.

[42] FBench. (2008, 10/11/2008). *Open Source Function Block Engineering Tool*. Available: http://oooneida-fbench.sourceforge.net/

[43] A. A. Jamshidifar, A. Afshar, and A. Salari, "FBLab, an IEC61499 based software tool for modeling of distributed control systems," *IEEE Conference on Control Applications*, pp. 1345-1349, 2003.

[44] G. Čengić, O. Ljungkrant, and K. Akesson, "Formal modeling of function block applications running in IEC 61499 execution runtime," *2006 IEEE Conference on Emerging Technologies & Factory Automation, Vols 1 -3*, pp. 918-925, 2006.

[45] ICSTriplex. (2009, 1/04). *ISaGRAF v. 5.0*. Available: www.isagraf.com

[46] nxtControl. (2010, 10/05). *nxtStudio*. Available: www.nxtcontrol.com

[47] V. Vyatkin, H.-M. Hanisch, P. H. Starke, and S. Roch, "Formalisms for verification of discrete control applications on example of IEC1499 function blocks," in *Verteilte Automatisierung (Distributed Automation)*, Magdeburg, Germany, 2000.

[48] L. Ferrarini and C. Veber, "Implementation approaches for the execution modle of IEC 61499 applications," in *2nd IEEE International Conference on Industrial Informatics (INDIN'04)*, Berlin, Germany, 2004, pp. 612-617.

[49] V. Vyatkin and V. Dubinin, "Sequential axiomatic model for execution of basic function blocks in IEC61499," *5th IEEE International Conference on Industrial Informatics*, pp. 1183-1188, 2007.

[50] V. Vyatkin, V. Dubinin, C. Veber, and L. Ferrarini, "Alternatives for execution semantics of IEC61499," *5th IEEE International Conference on Industrial Informatics*, pp. 1151-1156, 2007.

[51] C. Sunder, A. Zoitl, J. H. Christensen, V. Vyatkin, R. W. Brennan, A. Valentini, L. Ferrarini, T. Strasser, J. L. Martinez-Lastra, and F. Auinger, "Usability and Interoperability of IEC 61499 based distributed automation systems," in *IEEE International Conference on Industrial Informatics*, 2006, pp. 31-37.

[52] T. Strasser, A. Zoitl, J. H. Christensen, and C. Sunder, "Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems," *IEEE Transactions on Systems Man and Cybernetics, Part C-Applications and Reviews*, vol. 41, pp. 41-51, Jan 2011.

[53] G. Doukas and K. Thramboulidis, "Implementation Model Alternatives for IEC 61499 Function Block Networks," in *6th IEEE*

International Conference on Industrial Informatics, 2008, pp. 270-275.

[54] V. Vyatkin, "The IEC 61499 Standard and Its Semantics," *IEEE Industrial Electronics Magazine*, vol. 3, pp. 40-48, Dec 2009.

[55] G. Black and V. Vyatkin, "Intelligent Component-Based Automation of Baggage Handling Systems With IEC 61499," *IEEE Transactions on Automation Science and Engineering*, vol. 7, pp. 337-351, Apr 2010.

[56] J. Yan and Vyatkin V., "Distributed Execution and Cyber-Physical Design of Baggage Handling Automation with IEC 61499," in *9th International IEEE Conference on Industrial Informatics (INDIN'11)*, Lisbon, Portugal, 2011.

[57] G. Frey and T. Hussain, "Modeling techniques for distributed control systems based on the IEC 61499 standard - current approaches and open problems," in *Discrete Event Systems, 8th International Workshop on*, 2006, pp. 176-181.

[58] M. Bonfe and C. Fantuzzi, "Design and verification of mechatronic object-oriented models for industrial control systems," in *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, 2003, pp. 253-260 vol.2.

[59] V. Dubinin, V. Vyatkin, and T. Pfeiffer, "Engineering of validatable automation systems based on an extension of UML combined with function blocks of IEC 61499," *IEEE International Conference on Robotics and Automation (ICRA'06)*, pp. 3996-4001, 2005.

[60] K. Thramboulidis, "Model-integrated mechatronics-toward a new paradigm in the development of manufacturing systems," *Industrial Informatics, IEEE Transactions on*, vol. 1, pp. 54-61, 2005.

[61] S. Panjaitan and G. Frey, "Combination of UML modeling and the IEC 61499 function block concept for the development of distributed automation systems," *IEEE Conference on Emerging Technologies & Factory Automation*, pp. 449-456, 2006.

[62] A. Barji, N. Hagge, and B. Wagner, "Comparative study of using CNet, IEC 61499, and statecharts for behavioral models of real-time control applications," *IEEE Conference on Emerging Technologies & Factory Automation*, pp. 304-311, 2006.

[63] M. Hirsch and H.-M. Hanisch, "Systemspezifikation mit SysML für eine Fertigungstechnische Laboranlage," in *Fachtagung zum Entwurf komplexer Automatisierungssysteme (EKA 08)*, Magdeburg, Germany 2008, pp. 23-34.

[64] M. Hirsch, *Systematic Design of Distributed Industrial Manufacturing Control Systems*. Berlin: Logos Verlag 2010.

[65] V. Vyatkin, H. M. Hanisch, C. Pang, and C.-H. Yang, "Application of Closed-Loop Modelling in Integrated Component Design and Validation of Manufacturing Automation," *IEEE Transactions on Systems, Machine and Cybernetics - Part C*, vol. 39, pp. 17-28, 2009.

[66] J. L. Martinez Lastra and O. J. Lopez Orozco, "Semantic Extension for Automation Objects," in *IEEE International Conference on Industrial Informatics*, 2006, pp. 892-897.

[67] R. W. Brennan, L. Ferrarini, J. L. M. Lastra, and D. V. Vyatkin, "Automation objects: enabling embedded intelligence in real-time mechatronic systems," *International Journal of Manufacturing Research*, vol. 1, pp. 379-381, 2006.

[68] V. Vyatkin, H. M. Hanisch, P. Cheng, and Y. Chia-Han, "Closed-Loop Modeling in Future Automation System Engineering and Validation," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, pp. 17-28, 2009.

[69] D. Woll, "Will IEC 61499 trigger another automation system migration crisis?," *Hydrocarbon Processing*, vol. 86, pp. 15-15, Mar 2007.

[70] T. Hussain and G. Frey, "Migration of a PLC controller to an IEC 61499 compliant distributed control system: Hands-on experiences," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 3984-3989.

[71] C. Gerber, H. M. Hanisch, and S. Ebbinghaus, "From IEC 61131 to IEC 61499 for distributed systems: a case study," *EURASIP J. Embedded Syst.*, pp. 1-8, 2008.

[72] M. Wenger, A. Zoitl, C. Sunder, and H. Steininger, "Transformation of IEC 61131-3 to IEC 61499 based on a model driven development approach," *2009 7th Ieee International Conference on Industrial Informatics, Vols 1 and 2*, pp. 715-720, 2009.

[73] W. Dai and V. Vyatkin, "A Case Study on Migration from IEC 61131 PLC to IEC 61499 Function Block Control," *7th IEEE International Conference on Industrial Informatics*, pp. 79-84, 2009.

[74] G. Shaw, P. Roop, and Z. Salcic, "Reengineering of IEC 61131 into IEC 61499 Function Blocks," in *8th IEEE International Conference on Industrial Informatics*, 2010.

[75] A. Zoitl, T. Strasser, C. Sunder, and T. Baier, "Is IEC 61499 in Harmony with IEC 61131-3?," *IEEE Industrial Electronics Magazine*, vol. 3, pp. 49-55, Dec 2009.

[76] V. Vyatkin, J. H. Christensen, and J. L. M. Lastra, "OOONEIDA: an open, object-oriented knowledge economy for intelligent industrial automation," *Industrial Informatics, IEEE Transactions on*, vol. 1, pp. 4-17, 2005.

[77] J. Peltola, J. Christensen, S. Sierla, and K. Koskinen, "A migration path to IEC 61499 for the batch process industry," in *5th IEEE International Conference on Industrial Informatics*, 2007, pp. 811-816.

[78] D. Ivanova, I. Batchkova, S. Panjaitan, F. Wagner, and G. Frey, "Combining IEC 61499 and ISA S88 for Batch Control," in *13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09)*, Moscow, 2009.

[79] W. Lepuschitz and A. Zoitl, "Integration of a DCS based on IEC 61499 with Industrial Batch Management Systems," in *13th IFAC Symposium on Information Control Problems in Manufacturing*, Moscow, 2009.

[80] C. Pang and V. Vyatkin, "IEC 61499 function block implementation of Intelligent Mechatronic Component," in *8th IEEE Conference on Industrial Informatics (INDIN'10)*, Osaka, Japan, 2010, pp. 1124-1129.

[81] N. Higgins, V. Vyatkin, N. K. C. Nair, and K. Schwarz, "Distributed Power System Automation With IEC 61850, IEC 61499, and Intelligent Control," *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 41, pp. 81-92, Jan 2011.

[82] N. Hagge and B. Wagner, "Implementation Alternatives for the OMAC State Machines Using IEC 61499," *2008 Ieee International Conference on Emerging Technologies and Factory Automation, Proceedings*, pp. 215-220, 2008.

[83] M. Colla, T. Leidi, M. Kunt, and J. P. Thiran, "CEC Designer: Domain Specific Modelling for the Industrial Automation Based on the IEC 61499 Standard," *2008 IEEE International Conference on Emerging Technologies and Factory Automation, Proceedings*, pp. 192-199, 2008.

[84] S. Deen, *Agent-Based Manufacturing - Advances in the Holonic Approach, Springer 2003*: Springer, 2003.

[85] M. Fletcher and D. H. Norrie, "Realtime reconfiguration using an IEC 61499 operating system," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, 2001, pp. 985-991.

[86] T. Strasser, I. Muller, C. Sunder, O. Hummer, and H. Uhrmann, "Modeling of reconfiguration control applications based on the IEC 61499 reference model for Industrial Process Measurement and Control Systems," in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006)*, 2006, pp. 127-132.

[87] A. Zoitl, G. Grabmair, F. Auinger, and C. Sunder, "Executing real-time constrained control applications modelled in IEC 61499 with respect to dynamic reconfiguration," in *Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on*, 2005, pp. 62-67.

[88] M. N. Rooker, C. Sünder, T. Strasser, A. Zoitl, O. Hummer, and G. Ebenhofer, "Zero Downtime Reconfiguration of Distributed Automation Systems: The εCEDAC Approach," in *3rd Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems*, Regensburg, 2007, pp. 326-337.

[89] R. W. Brennan, M. Fletcher, and D. H. Norrie, "An agent-based approach to reconfiguration of real-time distributed control systems," *Robotics and Automation, IEEE Transactions on*, vol. 18, pp. 444-451, 2002.

[90] Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 26, pp. 381-391, 2010.

[91] M. Khalgui and H. M. Hanisch, "Reconfiguration Protocol for Multi-Agent Control Software Architectures," *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 41, pp. 70-80, Jan 2011.

[92] X. Huang, "Distributed and reconfigurable control of lower level machines in automatic production line," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, 2010, pp. 2339-2344.

[93] M. Budhia, V. Vyatkin, and G. A. Covic, "Powering flexible manufacturing systems with intelligent contact-less power transfer," in *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* Daejeon, 2008.

[94] W. Lepuschitz, A. Zoitl, M. Valle, and M. Merdan, "Toward Self-Reconfiguration of Manufacturing Systems Using Automation Agents," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, pp. 52-69, 2011.

[95] I. Hegny, O. Hummer, A. Zoitl, G. Koppensteiner, and M. Merdan, "Integrating Software Agents and IEC 61499 Realtime Control for Reconfigurable Distributed Manufacturing Systems," *2008 International Symposium on Industrial Embedded Systems*, pp. 249-252, 2008.

[96] X. M. Huang, "Intelligent and Reconfigurable Control of Automatic Production Line by Applying IEC61499 Function Blocks and Software Agent," in *IEEE International Conference on Mechatronics and Automation*, 2009, pp. 1481-1486.

[97] V. Vyatkin, H.-M. Hanisch, S. Karras, and X. Cai, "IEC61499 as an architectural framework to integrate formal models and methods in practical control engineering," in *Congress Electric Automation SPS/IPC/Drives*, K. Bender, Ed., ed Heidelberg, German, 2002, pp. 310-318.

[98] J. L. M. Lastra, L. Godinho, A. Lobov, and R. Tuokko, "An IEC 61499 application generator for scan-based industrial controllers," in *3rd IEEE International Conference on Industrial Informatics (INDIN '05)*, 2005, pp. 80-85.

[99] PROFACTOR. (2010). *Reconfigurable Robotic System*. Available: http://www.fordiac.org/25.0.html

[100] L. Ferrarini, M. Romano, and C. Veber, "Automatic Generation of AWL Code from IEC 61499 Applications," in *Industrial Informatics, 2006 IEEE International Conference on*, 2006, pp. 25-30.

[101] G. Polakow and M. Metzger, "Design and Implementation of LabVIEW-Based IEC61499 Compliant Device," *Holonic and Multi-Agent Systems for Manufacturing, Proceedings*, vol. 5696, pp. 183-192, 2009.

[102] D. O'Sullivan and D. Heffernan, "VHDL architecture for IEC 61499 function blocks," *Computers & Digital Techniques, IET*, vol. 4, pp. 515-524, 2010.

[103] F. Weehuizen and A. Zoitl, "Using the CIP protocol with IEC 61499 communication function blocks," *5th IEEE International Conference on Industrial Informatics*, vol. 1-3, pp. 261-265, 2007.

[104] V. Vyatkin, M. De Sousa, and A. Zoitl, "Communication aspects of IEC 61499 architecture," in *Industrial Electronics Handbook*, ed: Taylor & Francis, 2010.

[105] CAN-CiA. (2011). *Call for IEC 61499 experts*. Available: http://www.can-cia.org/index.php?id=1256

[106] C.-h. Yang and V. Vyatkin, "Model transformation between MATLAB simulink and Function Blocks," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, pp. 1130-1135.

[107] V. Vyatkin, G. Zhabelova, N. Higgins, M. Ulieru, K. Schwarz, and N. K. C. Nair, "Standards-enabled Smart Grid for the future Energy Web," in *1st IEEE and NIST Conference Innovative Smart Grid Technologies (ISGT)*, 2010, pp. 1-9.

[108] H.-M. Hanisch, M. Hirsch, D. Missal, S. Preusse, and C. Gerber, "One Decade of IEC 61499 Modeling and Verication - Results and Open Issues," in *Preprints of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, Moscow, Russia, 2009.

[109] G. Čengić and K. Akesson, "On Formal Analysis of IEC 61499 Applications, Part A: Modeling," *IEEE Transactions on Industrial Informatics*, vol. 6, pp. 136-144, May 2010.

[110] L. V. Allen, K. M. Goh, and D. M. Tilbury, "Closed-Loop Determinism for Non-Deterministic Environments: Verification for IEC 61499 Logic Controllers," in *2009 IEEE International Conference on Automation Science and Engineering*, 2009, pp. 1-6.

[111] P. Cheng and V. Vyatkin, "Automatic model generation of IEC 61499 function block using net condition/event systems," in *6th IEEE International Conference on Industrial Informatics (INDIN'08)*, 2008, pp. 1133-1138.

[112] C. Gerber, I. Ivanova-Vasileva, and H. M. Hanisch, "Formal modelling of IEC 61499 function blocks with integer-valued data types," *Control and Cybernetics*, vol. 39, pp. 197-231, 2010.

[113] C. Suender, V. Vyatkin, and A. Zoitl, "Formal Validation of Downtimeless System Evolution in Embedded Automation Controllers," *ACM Transactions on Embedded Control Systems*, 2011.

[114] P. Tait, "A path to industrial adoption of distributed control technology," in *IEEE Conference on Industrial Informatics (INDIN'05)*, Perth, Australia, 2005, pp. 86-91.

[115] M. Colla, A. Brusaferri, and E. Carpanzano, "Applying the IEC-61499 model to the shoe manufacturing sector," *IEEE Conference on Emerging Technologies & Factory Automation*, pp. 569-576, 2006.

[116] K. Thramboulidis, S. Sierla, N. Papakonstantinoul, and K. Koskinen, "An IEC 61499 based approach for distributed batch process control," *5th IEEE International Conference on Industrial Informatics*, pp. 177-182, 2007.

[117] C. C. Insaurralde, M. A. Seminario, J. F. Jimenez, and J. M. Giron-Sierra, "IEC 61499 model for avionics distributed fuel systems with networked embedded holonic controllers," in *IEEE Conference on Emerging Technologies & Factory Automation*, 2006, pp. 1128-1136.

[118] C. Pang, J. Yan, V. Vyatkin, and S. Jennings, "Distributed IEC 61499 Material Handling Control based on Time Synchronization with IEEE 1588," in *IEEE International Conference on Clock Synchronisation*, Munich, 2011.

[119] NxtControl. (2010, July). *Delta robot*. Available: http://www.nxtcontrol.com/no_cache/en/newsdetails/artikel/177/nxtcontrol-praesentiert-neue-version-auf-der-spsipcdrives-2010.html?cHash=c9fc250964864fa5bfb5e0b81896f252&sword_list%5B0%5D=demo

[120] IEC61499-4, "Function Blocks for Industrial Process Measurement and Control Systems, Part 4: Rules for Compliance Profiles," ed. Geneva: International Electrotechnical Commission, 2005.

[121] ITA, "IEC 61499 Compliance Profile for Feasibility Demonstrations," 2002.

[122] OOONEIDA, "Compliance Profile -- Execution Models of IEC 61499 Function Block Applications," 2009.

[123] "IEC 61499 Compliance Profile for CIP based communication function blocks," 2009.