Distributed Execution and Cyber-Physical Design of Baggage Handling Automation with IEC 61499

Jeffrey Yan and Valeriy V. Vyatkin, *Senior Member, IEEE*

***Abstract*— This paper proves feasibility of a fully distributed automation design of Baggage Handling Systems automation. The proposed design methodology leverages IEC 61499 Function Blocks as a supporting architecture. Each physical element of the BHS, such as a conveyor is represented by a Function Block that encapsulates functionality into a single re-usable module. The proposed solution demonstrates such benefits as flexibility, fault resilience and design re-use. A graph based representation of the system layout was used for rendering of the visualisation. The solution was prototyped using ISaGRAF – the first commercial implementation of IEC 61499 and tested on a network of 50 networked control nodes, each representing an embedded controller of a single conveyor or of a group of conveyors. The Ethernet communications framework was tested to confirm the appropriate levels of network utilisation and potential sources of delay were identified. This design approach can be regarded as cyber-physical since it naturally combines simulation models into the design life-cycle, providing ready to use simulation within the design framework.**

***Index Terms*— Distributed automation, component design, multi agent control, holonic control, IEC 61499, cyber-physical systems, baggage handling systems, simulation, ISaGRAF**

I. Introduction

The development of complex automation systems, such as airport Baggage Handling Systems (BHS), poses several major design challenges. The traditional automation systems engineering technologies, oriented on Programmable Logic Controllers (PLC), are of very low level of abstraction. They are not sufficiently scalable, do not provide system level verification and validation abilities, and are very much platform dependent despite some standardisation efforts.

Current mainstream design processes decouple the design of the physical system from software design. Thus, in BHS, the mechanical and electric parts are usually designed first, and then the software design follows with a little connection to the former. However many complex engineering systems, including BHS, now combine mechatronic components, computer hardware and software in such a way that reintegration of their design processes is highly beneficial.

IEC 61499 Function Blocks is an open standard that provides a reference architecture specifically aimed at distributed systems design. Benefits of the new standard for application development include portability, interoperability, re-configurability and distribution [3-4]. However, widespread acceptance of the new standard is yet to be achieved due to initial switching costs [5] and staff accustomed to older standards. The standard was proposed by the IEC to extend the older IEC 61131-3 Function Block standard for programming PLC systems. It defines a re-usable software mechanism called a Function Block (FB) used to encapsulate the functionality of a mechatronic unit. It is possible to encapsulate PLC logic into an FB and there are ways to migrate from the IEC 61131-3 PLC architecture to IEC 61499 [6], [7], [8-9]. A common problem during system migration is determining how the hierarchical system architecture is mapped to distributed nodes. Thus the simplest systems to migrate are those that have an inherently modular physical structure. Thus IEC 61499 may provide an excellent supporting architecture towards the new concept of Cyber Physical Systems (CPS) design [1] which has been coined to provide a design paradigm adequate to the current challenges. CPS vision is based on tight coupling and coordination between the physical and computational elements. Applications for CPSs have the potential to dwarf the information technology revolution of the past century. CPS research was placed in the top of the list for networking and information research in 2007 by the US President's Council of advisors [2].

Early implementations of system distribution involved splitting the control program and writing joining code to knit the smaller components into a complete system [10]. More modern techniques describe the notion of an intelligent, autonomous, goal-oriented agent in a multi-agent system [11]. Holonic Manufacturing Systems (HMS) [12] research has aimed to provide methodologies and architectures for development of multi-agent systems. Varying development platforms have been suggested including solutions based on the Function Block technology [13-14].

In this paper we present an extension of the CPS design applied in complex automation systems such as BHS. The layout of the plant serves as a primary source to describe the architecture of the control code. Furthermore, control code is adaptable and this is achieved by using agent concepts to modularise each controller. Each agent is executed directly in the physical device. However, implementation of this concept raises a number of research challenges including how to compose intelligent agents using the IEC 61499 standard as well as integrating typical BHS control functions. Additionally there is the question of performance and cost efficiency of

Manuscript received February, 4th, 2011

available platforms.

The paper is structured as follows. Section II introduces the concept of modularised BHS design and the proposed approach of designing a solution. Section III introduces the architecture of the system basic building blocks and their further partition into the Model-View-Control paradigm. Section IV introduces an automated layout format and the development of a visualisation application. Section V provides an analysis of the proposed distributed BHS control impact on network performance with measurements of network utilisation and event delay. Finally, Section VI concludes this research with a brief summary of the work along with potential future extensions.

II. Modular Baggage Handling System Design

The use of IEC 61499 for distributed multi-agent control of BHS was proposed in [15]. In [16] it was investigated and prototyped, using FBDK [17] as implementation environment. The work in [16], while providing an initial implementation of intelligent distributed BHS control, raises concerns in terms of scalability and ease of design. The prototyping in [16] was done on a Java-based function block execution platform FBRT whose performance and determinism cannot be regarded as industrial-grade. Furthermore the design process applied was far from seamless.

The design process proposed in [16] applies the Model-View-Controller design pattern and provides additional modularity upon IEC61499's already inherent modularity. The term "Model-View-Controller (MVC) architecture" was originally coined in the context of object-oriented language Smalltalk-80 [18], where it was used to define a software design paradigm that separates the application logic (Model), the display of current application state (View) and user input (Controller). It has since seen widespread adoption among a multitude of programming frameworks including IEC 61499 FBs [19]. In the BHS context, the model component represents a simulation of a physical conveyor and can be used to test the controller in absence of the physical plant. The view component aggregates useful information from the model or plant and renders it on a remote display.

Cyber Physical Systems design may have synergies with the MVC design pattern. Design flow could start at the physical plant layout which would describe the architecture of each component in the MVC triad. Offline simulation could then be applied before deployment of control code to the plant.

The goal of this work is to develop an efficient design method for BHS automation with much reduced design and validation effort along with much improved robustness and adaptability of the resulting BHS, as compared to the state-of-the-art automation methods. We propose a cyber-physical design method with ambition to obtain control code automatically simply from reproducing the system layout (Figure 1). We call this approach "cyber-physical" due to the properties of the physical part (layout, connections, dynamics) being essentially used in automatic generation of the cyber part (control, communication and visualisation code).

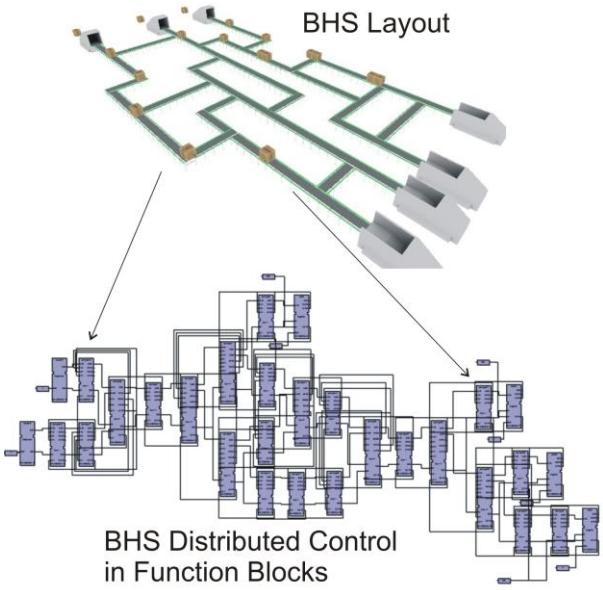

Figure 1. Cyber-Physical BHS control design concept: from layout to decentralized control and distributed deployment.

The distributed control code will consist of intelligent agents' instances and explicit event and data flow between them. Each agent type will be implemented as a function block type following IEC 61499. The intelligent agent is structured internally following the MVC architecture and is implemented as a function block. ISaGRAF a commercial off-the-shelf implementation of IEC 61499 is used for development. It supports the older IEC 61131-3 standard for programming PLCs and has recently added support for IEC 61499. Compared to other IEC 61499 Integrated Development Environments (IDEs) ISaGRAF has several distinct features, some of which are outlined in [20], however the major benefit is the ability of seamless application distribution. The developer can simply specify which devices each FB will reside on and ISaGRAF will automatically insert the necessary communication.

The hardware test-bed used in this work consisted of 48 Netburner devices (core of the ISaGRAF demo-kit) which as a prototype of future conveyor embedded controllers as shown in Figure 2. Each controller runs the μC/OS-II Real-Time operating system along with ISaGRAF runtime for execution of traditional PLC and IEC 61499 distributed programs. These controllers represent a prototypical example of an intelligent lightweight module which could be integrated into every conveyor or motor drive.

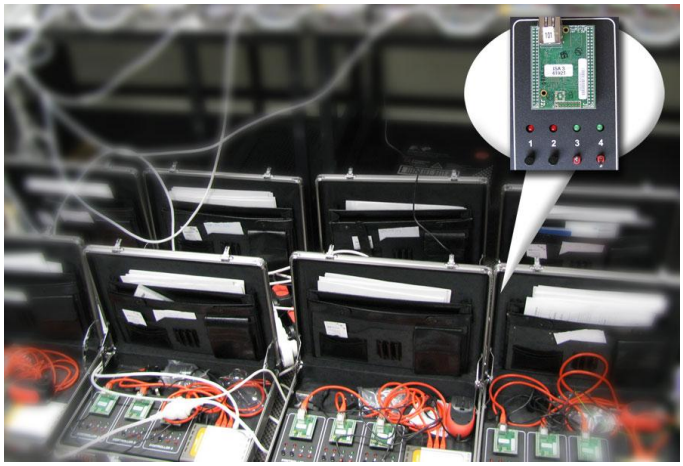

Figure 2. Network of 48 conveyor embedded controllers (Netburner) - the core of the distributed BHS testbed

III. Multi-Agent BHS Design based on IEC 61499

Airport BHSs usually consist of a combination of conveyors, diverters, beacons, X-ray and other scanners with conveyors making up the majority of the functionality. Conveyors are arranged in layouts according to airport requirements; however, the layout may be subject to considerable changes during the system's lifecycle to account for new bag routes, changes to equipment or malfunctions. In the proposed architecture, a conveyor FB agent serves as a basic building block for the layout based BHS control code design. Each conveyor FB contains control code and simulation model code encapsulated into a single module structured according to the MVC architecture. The View part of the conveyor agent communicates with a standalone visualization application that is rendering the BHS.

A. Conveyor Model Component

The Model component represents an accurate behavioural model of the plant (including the conveyor belt, sensors and actuators) or an interface to its sensors/actuators (selectable). The model reflects low level mechanical functions such as conveyor belt movement and bag movement. It allows the ability to inject bags, move bags and finally extract bags. Corresponding photo-eye sensors are triggered according to the position of bags on the conveyor. Control signals triggering the running of the motor are given in the form of Boolean signals representing photo-eye state.

B. Conveyor Controller Component

The conveyor controller handled low level mechanical functionality as well as providing intelligent behaviour such as merging, diverting and bag tracking. While a traditional BHS may have a centralised database, a truly distributed system is composed of re-usable software modules thus distributed tracking methods must be employed. When a bag is passed from one conveyor to the next, a record of its bag-data is passed along with it.

The high level functionality of the controller was further separated logically using a notion of a virtual-conveyor. In the previous work [16], a single parameterised controller was used per conveyor module. However the high level control here was further split into separate virtual modules referred to here as virtual-conveyors. Each virtual-conveyor represents the control for a simple conveyor section consisting of no more than a single merge point, a single divert point and a single photo-eye sensor. The controller for a more complex conveyor section would then be an aggregation of multiple virtual-conveyor controllers. The conveyor section in Figure 3 is an example with two photo-eye sensors, and a diverter following each photo-eye.

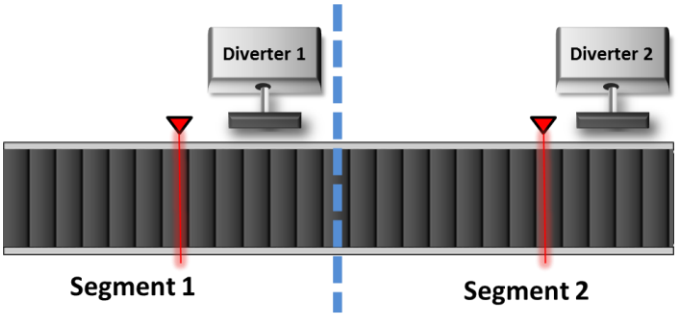

Figure 3. Example Conveyor Section with two photo-eye sensors and two diverters and an illustrated logical split position

In Figure 3, control code for each section Segment1 and Segment2 would be almost identical and each would handle bags on their own segments. The logical split is taken such that the first and second segments contain a single photo-eye and diverter.

Code structure will consist of a virtual-controller per segment which handles bags only on the corresponding segment. This is shown in the FB diagram in Figure 4. A virtual-conveyor controller is instantiated as many times as necessary within a composite FB to compose the control code for the overall conveyor section.

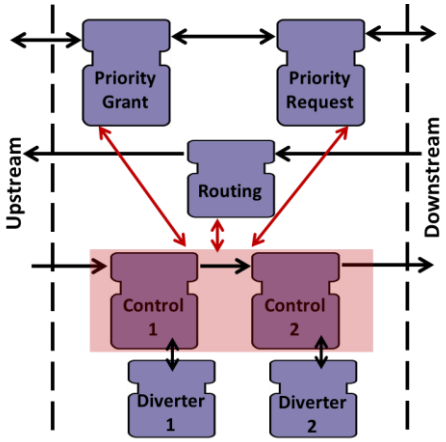

Figure 4. Internal structure of corresponding control code FBs for a conveyor with 2 photo-eye sensors and 2 diverters

C. Conveyor View Component

The view component of the MVC triad is responsible for providing an accurate representation of the system state, possibly to an external visualization application. We have developed such an application to render the state of the BHS

on a local or remote computer. Thus the view component had the sole purpose of providing data to the visualisation by conveying them to the OPC server.

D. Dynamic Path Planning

Routing bags to their correct destinations is an essential objective of any BHS. In traditional BHSs the path planning is done in a separate application that knows the entire BHS topology. However in the case of distributed BHS, each conveyor only communicates to its immediate neighbours, either downstream or upstream. Besides, we assume that due to possible faults the set of available active conveyors is changing over the time. Our distributed BHS FB system aims to provide baggage routing capabilities which are equivalent to those in a centralised system.

The base of our routing solution is the distributed Bellman-Ford (DBF) algorithm with a simple example illustrated in Figure 5. At the time t=0 the routing table for Node-A contains only distance metrics to its direct neighbours Node-B and Node-C. On the next iteration of DBF, Node-A will receive routing tables from Node-B and Node-C. New information received during this iteration is highlighted and now Node-A has information about forwarding to Node-D and Node-E through an intermediate node. A node will broadcast to its neighbours if it has any new information about shortest paths.

t=0

from A	via A	via B	via C	via D	via E
to A					
to B		7			
to C			1		
to D					
to E					

t=1

from A	via A	via B	via C	via D	via E
to A					
to B		7	9		
to C		15	1		
to D		8			
to E			3		

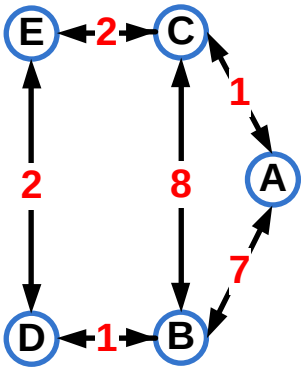

Figure 5. Network of 5 routers with distance metrics and the routing tables for Node A at time t=0 and t=1

IV. Layout-Based Visualization and Control Design

Implementation of the control system in IEC 61499 FBs is the first step towards a Cyber Physical design approach. If the designer only has access to a CAD layout diagram of the plant and the necessary FBs for each component then this already significantly eases the design process. However, rather than manually specifying how to draw the BHS layout to the visualisation, an implementation independent meta-model format describing BHS could assist us in rendering the BHS for on-screen visualisation and even generating the FB system.

In this work the system is represented as an attributed graph and stored in an XML file. An example of a visual rendering of this XML is in Figure 6. The program Attributed Graph Grammar System (AGG) was used for the drawing of this layout. Conveyors represent nodes on this graph and the connections between conveyors represent the edges.

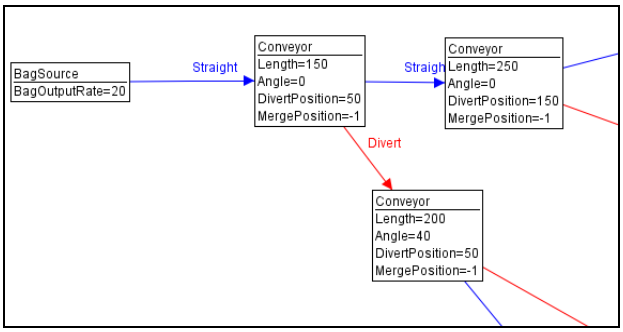

Figure 6. Graph representation of a small BHS section with a cross-section of conveyors

Using this attributed graph format a custom made visualization application (Figure 7) was able to render the BHS on screen and simultaneously set up an OPC server with the current configuration. ISaGRAF provides an OLE for Process Control (OPC) server for the communication of real-time data between the control devices and other applications, such as SCADA and Human-Machine Interface. Fault tolerance mechanisms within the conveyor controller indicate to the visualization when a conveyor has become faulty or has gone offline and these conveyors are subsequently highlighted in the visualization.

The same graph format could also be used to generate the FB system for immediate deployment however due to time constraints and the closed nature of ISaGRAF file formats this was not attempted in this work. Many industrial systems could potentially be represented by an attributed graph format. Furthermore another method in which to derive the FB system from the graph model would be to apply transformation techniques which are an integral function of the AGG software.

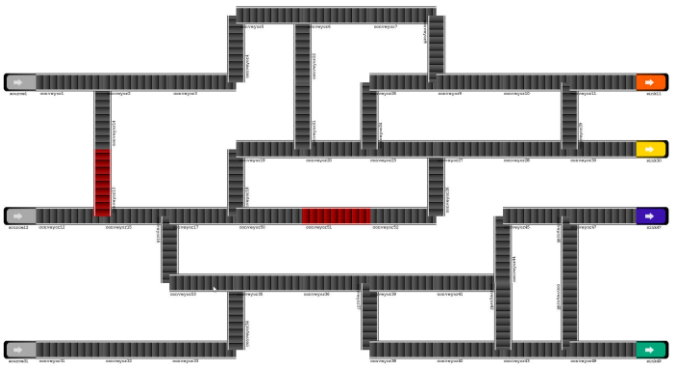

Figure 7. Visualization of a BHS configuration with 53 conveyors. The control is distributed among 27 ISaGRAF configurations (control nodes). Fault tolerance mechanisms have indicated to the visualization that some conveyors are faulty or are offline, shown in red.

V. Testing and Network Analysis

Due to the unpredictable nature of networks it is difficult to

guarantee performance metrics of distributed systems compared to using a centralized controller. Since IEC 61499 is a relatively new standard, not much is known about its performance characteristics. The previous work [16] involving distributed BHS control did not include any form of testing. Since IEC 61499 provides some new concepts for the majority of OEMs and system integrators, some preliminary performance analysis may provide further verification that control distribution is a viable solution.

Figure 8 shows a test-bed that consists of a real conveyor loop with feed in and feed out conveyors, which is considered to be a part of an airport BHS. The rest of the conveyors are simulated. The real part of the test-bed is equipped with Variable Speed Drives and WAGO control devices, the "virtual" simulated part is executed on the Netburner nodes (Figure 2).

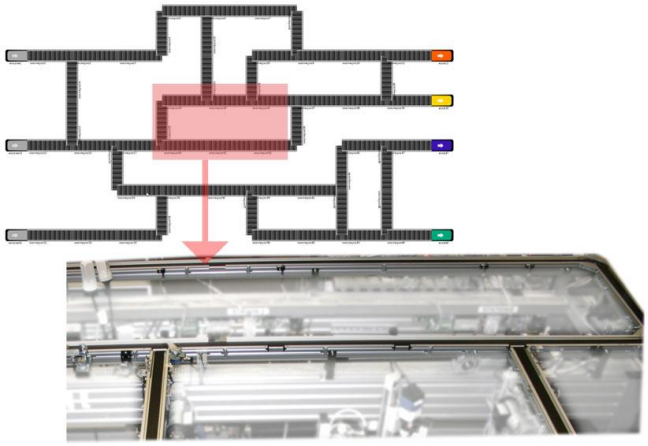

Figure 8. Test-bed combining a real conveyor loop with "virtual airport"

Since the NetBurner controllers did not have I/O, performance tests were executed and results recorded using FB applications integrated with the BHS logic. Network traffic analysis and event transmission delay tests were carried out.

A. Network Utilization Results

The BHS system in Figure 7 was used as a test case. It contains 53 conveyor Function Blocks distributed over 27 NetBurner controllers. Figure 9 shows the network utilisation of three controllers in the system. The results show that the three controllers are far from saturating the bandwidth that 100Mbit/s Ethernet can provide however this is to be expected for transmitting groups of primitive data types over Ethernet. In further testing, investigation of bursts of network traffic may further help to determine if there are any performance bottlenecks in the system.

The maximum dimension of BHS system tested in our design framework so far includes around 100 conveyors with controls distributed across 48 control nodes.

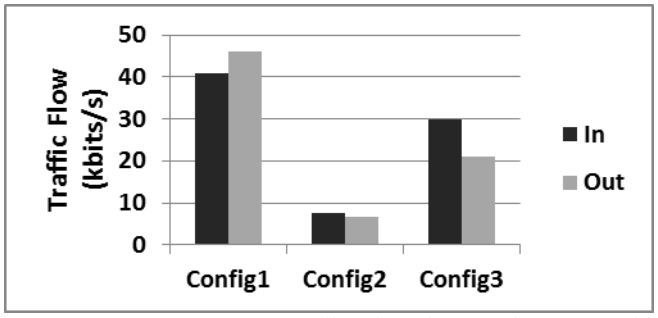

Figure 9. Average Load network traffic of Config1, Config2 and Config3 which represent the first 3 conveyors starting from the top-most bag-source

B. Event Transmission Delays

Event transmission delay can give the developer insight into system reaction time when the distributed system is under some form of load. An application to measure time between event transmission and receipt was run within the Config1 and Config2 controllers. Figure 10 shows the results for the delay tests as represented in box-plots. Most of the results suggest an event transmission delay of approximately 50ms.

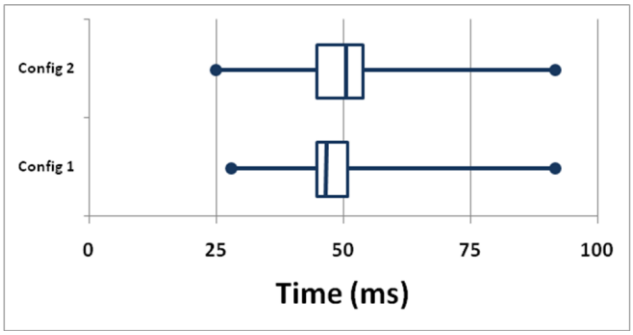

Figure 10. Event transmission delays between two configurations.

The average PLC cycle time was recorded to be 5.36ms and 5.71ms for Config1 and Config2. The fact that the delay is significantly larger than the average cycle time indicates that the code execution may either be experiencing a longer cycle time during event transmissions, or that the network delay is not negligible. This result should be further investigated.

VI. CONCLUSION

In the reported research we have fully confirmed the feasibility of implementing BHS control in a purely distributed way. The approach involved developing a conveyor FB which encapsulated typical conveyor functionality into an autonomous module. Focus was placed on implementing a cyber-physical methodology for the design and implementation of the BHS control and visualization. The approach further advances BHS design towards a goal of simple reconfiguration by way of relating computational elements (FBs) with physical elements (conveyors).

Designing the system involved connecting conveyor FBs to mirror the layout of the physical BHS system. However the visualization application subsequently developed used a attributed graph format drawn in AGG and could potentially be used for generation of the FB network. Integration of

industry standard design exchange formats such as CAD would be the next step in easing the design process for large distributed control systems.

The ISaGRAF implementation of the IEC 61499 standard allowed FBs to run on targeted hardware with simple user specification. A single FB could be assigned to run on a single controller or multiple FBs assigned to run on the same controller. Simulation of the BHS system was done on both centralized and distributed hardware configurations.

Future work may include investigation of automated generation of FB systems from a meta-model representation of the system. If the control code already exists as modularized FBs then hiding these from the system designer and only programming with visual techniques would be ideal. Additionally more detailed benchmarking is required to accurately compare a distributed BHS control solution to traditional centralized solutions.

ACKNOWLEDGMENTS

The authors thank **Glidepath**, a New Zealand company specialising in the area of Baggage Handling System Integration and Control. Glidepath provided the motivating details of the baggage handling automation. William Dai in particular provided valuable discussions. The authors are grateful to ISaGRAF for providing valuable product maintenance during this research.

VII. REFERENCES

[1] E. A. Lee, "Cyber Physical Systems: Design Challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, 2008, pp. 363-369.

[2] "Cyber-Physical Systems: Executive Summary," in *Cyber-Physical Systems Summit*, St. Louis, Missouri, 2008.

[3] C. Sunder, *et al.*, "Usability and Interoperability of IEC 61499 based distributed automation systems," in *Industrial Informatics, 2006 IEEE International Conference on*, 2006, pp. 31-37.

[4] J. H. Christensen, "Design patterns for system engineering with IEC 61499," in *Conference Verteilte Automatisierung*, Magdeburg, Germany, 2000, pp. 63-71.

[5] K. Zhu, *et al.*, "Migration to Open –Standard Interorganizational Systems: Network Effects, Switching Costs, and Path Dependency," *MIS Quarterly*, vol. 30, pp. 515-538, 2006.

[6] T. Hussain and G. Frey, "Migration of a PLC Controller to an IEC 61499 Compliant Distributed Control System: Hands-on Experiences," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 3984-3989.

[7] J. Peltola, *et al.*, "A Migration Path to IEC 61499 for the Batch Process Industry," in *Industrial Informatics, 2007 5th IEEE International Conference on*, 2007, pp. 811-816.

[8] W. Dai and V. Vyatkin, "Redesign Distributed IEC 61131-3 PLC System in IEC 61499 Function Blocks," presented at the IEEE International Conference on Emerging Technologies and Factory Automation, Bilbao, Spain, 2010.

[9] W. Dai and V. Vyatkin, "A Case Study on Migration from IEC 61131 PLC to IEC 61499 Function Block Control," presented at the 7th International IEEE Conference on Industrial Informatics (INDIN'09), Cardiff, 2009.

[10] K. H. Hall, *et al.*, "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.

[11] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2 ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[12] P. Valckenaers and H. Van Brussel, "Holonic Manufacturing Execution Systems," *CIRP Annals*, pp. 427-432, 2005.

[13] J. Christensen, "HMS/FB Architecture and its Implementation," in *Agent Based Manufacturing. Advances in the Holonic Approach*, M. Deen, Ed., ed: Springer, 2003.

[14] M. Fletcher, *et al.*, "An open architecture for holonic cooperation and autonomy," in *Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on*, 2000, pp. 224-230.

[15] V. Vyatkin, *et al.*, "Information Infrastructure of Intelligent Machines based on IEC61499 Architecture," *IEEE Industrial Electronics Magazine*, vol. 1, pp. 17-29, 2007.

[16] G. Black and D. V. Vyatkin, "Intelligent Component-based Automation of Baggage Handling Systems with IEC 61499," *IEEE Transactions on Automation Science and Engineering*, vol. 6, p. in print, 2009.

[17] HOLOBLOC., "Function Block Development Kit," ed, 2008. http://www.holobloc.com/.

[18] P. D. Steve Burbeck, "Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)," ParcPlace Systems, Inc.1987.

[19] J. H. Christensen, "IEC 61499 Architecture, Engineering Methodologies and Software Tools," presented at the Proceedings of the IFIP TC5/WG5.3 Fifth IFIP/IEEE International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services: Knowledge and Technology Integration in Production and Services: Balancing Knowledge in Product and Service Life Cycle, 2002.

[20] V. Vyatkin and J. Chouinard, "On Comparisons of the ISaGRAF implementation of IEC 61499 with FBDK and other implementations," presented at the IEEE International Conference on Industrial Informatics, 2008.