# DESIGN AND IMPLEMENTATION OF MANUFACTURING AUTOMATION SYSTEM THROUGH INTERNATIONAL STANDARDS

**E. Carpanzano, A. Cataldo, M. Donà**

*Institute of Industrial Technologies and Automation, National Research Council, Italy*
*e-mail : e.carpanzano, a.cataldo. m.dona@itia.cnr.it*

Abstract: Innovative development methodologies, standards and tools for the structured design of complex automation systems are very important to configure interoperable, portable and scalable automation systems for next generation manufacturing systems. In the present work a development methodology is applied with particular reference to the automation system functional specification, which is based on emerging international standards and exploits object-oriented concepts. An application example is also addressed throughout the paper to highlight the benefits of the proposed framework. In particular, the development of automation functionalities for an highly innovative shoe manufacturing plant and regarding test is discussed. 

Keywords: Automation, Development Methodology, IEC Standards, Object Orientation, Pilot Plant.

## 1. INTRODUCTION

Reliable and agile automation systems are a crucial point for competitiveness of modern manufacturing systems (Bossi, et al., 2003). Interoperability, portability and scalability of developed automation solutions are also fundamental elements to reduce the costs and times needed to design and realize a new production systems, or to modify an existent one. In such a context, the definition of development methodologies that support the structured design and testing of the whole automation system of a manufacturing plant is mandatory (Carpanzano and Ballarino, 2002). In the present work a framework for the design of control and supervision systems of modern manufacturing systems is discussed, in particular the automation system functional design is highlighted. One of the key elements of the proposed approach is the adoption of a structured design methodology (Ferrarini and Carpanzano, 2002). This is based on formal reference models, based on the object-oriented paradigm (Maffezzoni, et al., 1999), and compliant with IEC international standards. In particular, a functional automation system model is proposed which has been derived after analyzing the IEC 61499 standard (Standard IEC 61499, 2001). Moreover, the SFC (Sequential Functional Chart) formalism, included in the IEC 61131 standard (Standard IEC 61131, 1993), is exploited to develop and implement the automation software.

The paper is organized as follows. The design cycle model of the adopted development methodology is introduced in section 2. Then, the reference models and standards used for the development of the automation system are outlined in section 3. In section 4 the design of the automation system of an area of a manufacturing plant is described as an illustrative example, highlighting the advantages of the presented approach. Finally, concluding remarks are drawn (section 5).

## 2. PHASES OF THE DESIGN CYCLE MODEL

A structured development methodology for the design of industrial automation systems is based on a design cycle model (DCM) that establishes the order of the involved development phases. In the present work the DCM described in (Carpanzano and Ballarino, 2002) is used, such a DCM is based on the following phases.

A. *System definition*: the process to be automated is described, and the activities to be performed and the objectives of the automation system are defined.
B. *Automation system specification*: the tasks and the essential functions of the supervision and control system are defined. In order to meet the objectives of unification, reusability, and traceability, the automation system shall observe a common structuring scheme consistent through the subsequent phases of the DCM. Thus, the system can be structured into a hierarchy of different layers (plant, area, cell, unit, control system devices), based upon a successive decomposition of top-level control objectives into elementary control actions.
C. Automation system design: the supervision and control system are developed through suitable formal reference models and standards for both the architectural and the functional design.
D. *Automation system implementation and testing*: the automation software is developed and it is checked if the designed solutions meet the requirements. This may be achieved by means of suitable simulation methods.
E. *System integration and testing*: it is checked if the automation system works on its devices, and if all the system's requirements are met.

## 3. REFERENCE MODELS AND IEC STANDARDS

In the present section the adopted reference models and international standards for the design of the automation system architecture, functions and algorithms are introduced in sections 3.1, 3.2 and 3.3, respectively. In particular, the proposed models are based on basic object-oriented modelling concepts (Maffezzoni, et al., 1999) like:
- *Modularity*: a complex automation system is divided into smaller and more manageable modules;
- *Hierarchy*: the modules are structured on different hierarchic levels, according to a top-down functional decomposition approach;
- *Encapsulation*: every module hides its internal algorithms and variables to the other components of the system;
- *Aggregation*: the main modules of the system architecture are obtained through the aggregation of simpler submodules;
- *Inheritance*: specialized components can be derived from more general ones.

Major benefits deriving from the adoption of the proposed models are that the definition of complex automation systems is extremely simplified, reusability is improved, and re-usability is strongly enhanced.

### *3.1 Model for the automation system architecture.*

In order to define the reference model for the automation system architecture, different international standards for the representation of open control systems architectures have been taken into account. Specifically, the main features of the OSACA, OMAC and OSEC standards have been considered. Even if such standards are not compliant with each other, a new reference model for the automation system architecture is proposed in (Carpanzano, *et al.*, 2002) which exploits their common aspects. Such a model is sketched in figure 1, where the basic modules are illustrated, i.e.: *HMI and Configuration Tools, Logic Control, Motion Planning and Axis Control*. Such modules are further detailed in (Carpanzano et al., 2002).

### *3.2 Model for the functional design.*

The IEC 61499 standard, which defines function blocks for industrial process measurement and control systems, is here introduced to develop detailed functional models of the automation system software, that define also the distribution of the supervision and control functions on the physical devices (Schoop and Strelzoff, 1996), (Standard IEC 61499, 2001). The IEC 61499 standard is based on a fundamental module, the function block (FB), which represents a functional unit of software, associated to a hardware resource of the control system. As shown in figure 2, a FB instance is characterised by: its type name and instance name, sets of event inputs/outputs and data inputs/outputs, internal data, an execution control chart (ECC) consisting of states, transitions and actions, which invokes the execution of algorithms in response to event inputs; and a set of algorithms, associated with the ECC states.

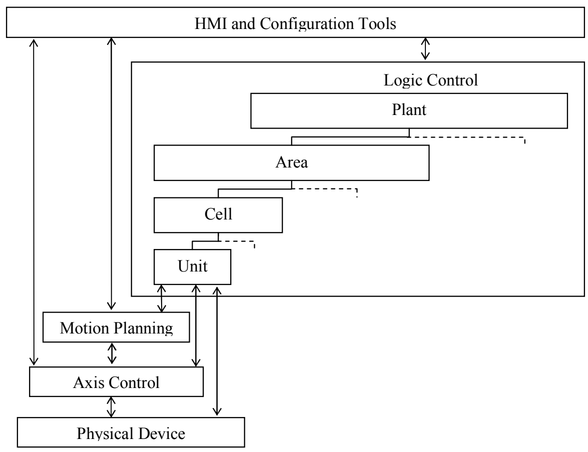


Fig. 1. Reference model for the automation system architecture.

The execution of algorithms is invoked by the ECC (which is basically a Moore automaton) of a FB instance, in response to event inputs. When the execution of an algorithm is scheduled, the needed input and internal data values are read and new values for output and internal data may be computed. Furthermore, upon completion of execution of an algorithm, the execution control part generates zero or more event outputs as appropriate. By properly connecting more FBs an application is defined.
As regards the configuration, an application can be distributed among several control system devices. A device uses the causal relationship specified by the application to determine the appropriate responses to events. Furthermore, in the IEC 61499 standard a

resource is considered to be a logical subdivision within the software (and possibly hardware) structure of a device, which has independent control of its operations. Each FB instance is associated to one single resource.
With the given definitions the architecture of a manufacturing automation system can be modelled as a collection of devices, divided in resources, interconnected and communicating with one another by means of one or more communication networks, while the functions performed by such a system are modelled as applications.

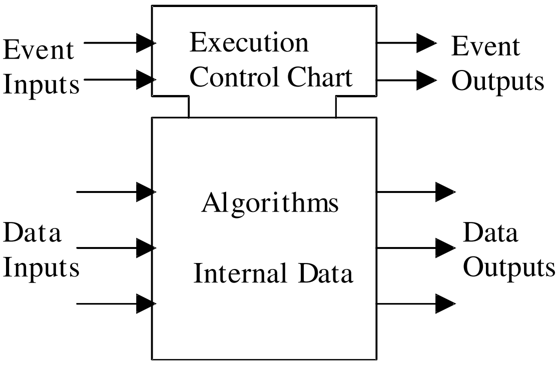


Fig. 2. Functional block model.

*3.3 Graphical programming language: Sequential Functional Chart.*

The Sequential Functional Chart (SFC) formalism, formerly also called Grafcet, was defined starting from Petri nets, so as to define a new graphical discrete event model suitable to describe logic control systems. Thus, there are many similarities in the structures and in the evolution rules of Petri nets (PNs) and SFCs (Renè, 1995). The SFC formalism has also been included in the IEC 61131 part 3 standard in 1993 as one of the five standard programming languages for PLCs (Lewis, 1998).
A SFC can be formally defined as a 4-tuple $SFC = (S, T, F, M_0)$,
where: $S = \{s_1, s_2, s_3 \cdots, s_m\}$ is a finite set of steps; $T = \{t_1, t_2, t_3, \cdots, t_n\}$ is a finite set of transitions; $F = (S \times T) \cup (T \times S)$ are sets of connections from steps to transitions and from transitions to steps; $M_0 : S \rightarrow \{0,1,2,3,\cdots\}$ is the initial marking; $S \cap T = \phi$ and $S \cup T \neq \phi$.
Because of the similarities between PNs and SFCs the Reachability, Liveness and Reversibility properties definitions for Petri nets can be directly applied also to the SFC formalism (Bossi, et al., 2003). Actually, an important issue in designing a manufacturing automation system is whether the system can reach a specific state or not, therefore the reachability property is of interest in the study of such systems. Furthermore, liveness guarantees deadlock-free operation, no matter what firing sequence is chosen. While, reversibility implies that the system will have a cyclic behavior and will perform its functions repeatedly; it also characterizes the recoverability of the initial state from any state of the system.

## 4. DESIGN OF THE AUTOMATION SYSTEM FOR AN INNOVATIVE SHOE MANUFACTURING PLANT

In this section the presented framework is applied to the design of the automation system of an innovative shoe manufacturing plant owned by ITIA-CNR and shown in figure 3. For the sake of brevity a simplified version of a part of the IMS is considered, see Figure 4. In particular, the design dealt with is illustrated by going through the different phases of the DCM introduced in section 2.

Fig. 3. Real shoe manufacturing system.

*4.1 System definition.*

In the considered IMS an innovative transport line is used to move the semifinished shoes from a machining station to another one according to a predefined operation schedule. The innovative molecular structure of the transport line of the making department enhances the modularity, scalability, integrability and reconfigurability properties of the production system, so increasing the overall flexibility of the plant. The basic element of the molecular structure is the “Tern”, which is constituted by two rotating tables, called “Table” and “Island”, and by a rotating three arms manipulator. The Table is used to direct the semifinished shoes either to the next Tern or to the Island of the same Tern. Moreover, it moves backward the lasts flowing back towards the warehouse (the last is the object around which the semifinished shoe is built upon). The Island directs the semifinished shoes towards the different machining stations, laid around the Island itself. The transportations of the semifinished shoes and lasts between Tables and Islands are carried out by the manipulators.

*4.2 Automation system specification.*

To achieve the desired agility objectives, object-oriented concepts have been exploited for the control system development (Maffezzoni et al. 1999). In particular, the molecular line has been considered as a set of interacting Terns, each one with its own independent control system (Carpanzano and Cataldo, 2003). So, each Tern operates autonomously according to its boundary conditions, while interfaces among different Terns are represented by semifinished shoes and lasts exchanges. With reference to a generic single Tern, the possible movements that involve a generic semifinished shoe or last are depicted by means of arrows in Figure 4.

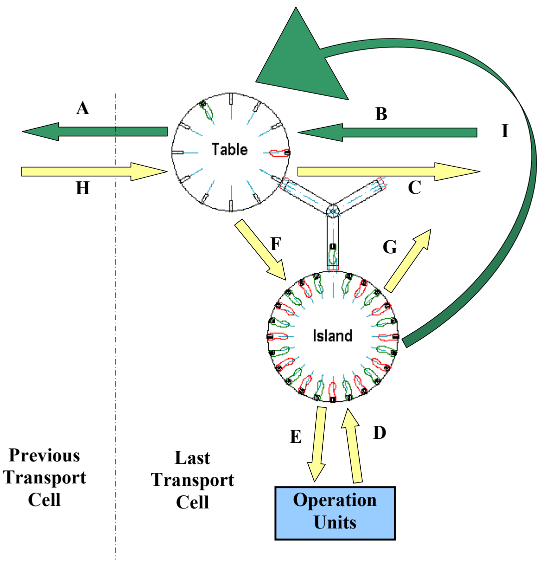


Fig. 4. Studied simplified system: Tern.

A semifinished shoe that arrives on the Table of a Tern (arrow H) and which has not to be machined by the Tern itself is directly moved to the next Tern (arrow C); otherwise, it is moved to the Island (arrow F), where it is properly machined (arrows D and E). For the sake of simplicity only one machining station has been here represented. As soon as the machining is over, the shoe is moved towards the next Tern (arrow G). Whenever a shoe is finished, it is removed from its last, and the last itself goes back to the warehouse by flowing back through the whole transport system. So, for a generic Tern, the last that must be stored in the warehouse comes from the adjacent Tern (arrow B) and goes back, through the Table, to the previous Tern (arrow A). Specifically, the lasts start their way back to the warehouse in the last Tern, once the shoe working process is over (arrow I). The proposed control strategy for a generic Tern is based on the following three basic assupmtions:

1. avoid deadlocks, i.e. avoid situations into which none
semifinished shoe or last can be moved;
2. favour the backward lasts flow respect to the forward one of the semifinished shoes;
3. favour the unloading of the Tern resources, i.e. of the Table, Island and Operation Stations.

Such heuristic rules have been defined to guarantee a correct and efficient use of the system resources. To implement the first assertion it has been decided to have always one slot free on a Table for the last backward movement toward the warehouse. Moreover, to implement the second and third assertions proper priorities have been assigned to the different possible operations illustrated with reference to Figure 4. Such priorities are represented in table 1. In the columns of the table the different operations have been listed. The presence of the character “x” in a cross between a column and a row means that the operation associated to the column has minor priority than the one associated to the row. Notice that the possibility to execute concurrent operations, e.g. operations A and H, is also considered in the table, e.g. operation AH. As a consequence, it is not significant to define a priority between operations A and H, since the concurrent operation AH can be carried out.

Table 1 Priority logic for the Tern

| | AH | A | BFG | BC | BF | BG | B | C |
|---|---|---|---|---|---|---|---|---|
| AH | - | X | X | X | X | X | X | X |
| A | - | - | X | X | X | X | X | X |
| BFG | - | - | - | X | X | X | X | X |
| BC | - | - | - | - | X | X | X | X |
| BF | - | - | - | - | - | X | X | X |
| BG | - | - | - | - | - | - | X | X |
| B | - | - | - | - | - | - | - | N.O. |
| C | - | - | - | - | - | - | N.O. | - |
| D | - | - | - | - | - | - | - | - |
| E | - | - | - | - | - | - | - | - |
| FG | - | - | - | - | - | - | - | - |
| F | - | - | - | - | - | - | N.O. | - |
| G | - | - | - | - | - | - | N.O. | - |
| H | - | N.O. | - | - | - | - | - | - |

### *4.3 Automation system design.*

As previously said, the control system has been designed by following a modular approach (Park et al. 2001). In particular, each Tern has an own control module, which communicates with the regarding Table, Island and Manipulator control modules, see figure 5. Moreover, each Tern control module is connected to the adjacent Terns control system modules, to coordinate the exchanges of semifinished shoes and lasts, and to the plant Supervisor that monitors the whole transport line.

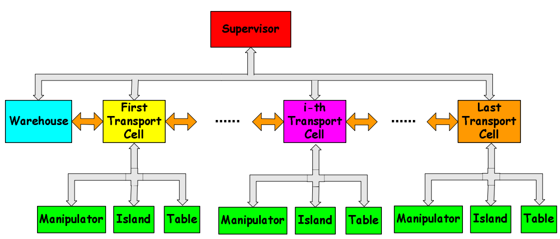


Fig. 5. Automation system architecture.

In order to design the functional behaviour of the control system of the tern, the standard IEC 61499 has been used (Lewis, 2001), see figure 6. In particular a set of functional blocks are connected in closed loop sequence, in which two kind of FB are used: one to manage the sequence execution of the FB functionalities, the second to describe the functional behaviour of the control system. In order to start the functional sequence, it is necessary to set high the event E_RUN, which flow thorough the FB E_MERGE (this FB set high the output event E_O if at least one of the input events E_I1 or E_I2 are set high). The FB which contain the control system functionalities (Control_Tern_1) is activated and, by means of the data input values, it starts the control algorithms, by generating opportune data output (control variables) and set high the regarding output event E_O. This event is split into different synchronized events E_O1, E_O2, …, E_On (from the FB E_SPLIT), so to start each of the basic FB contained in the composite FB "Rotating_Table_1", see Figure 7. The first basic FB "Rotation_Table_1" manages the slot position of the controlled Table of the transport line. The second and third FBs Pusher_T11 and Pusher_T12 describe the functionalities of the pushers used to move the semifinished shoes from a rotating table to another one. As soon as all the basic FBs have finished their algorithms, and then corresponding output events E_O1, E_O2 and E_O3 have been set high, then the FB E_REND set high the regarding output event E_O so to restart the FB sequence activation cycle.

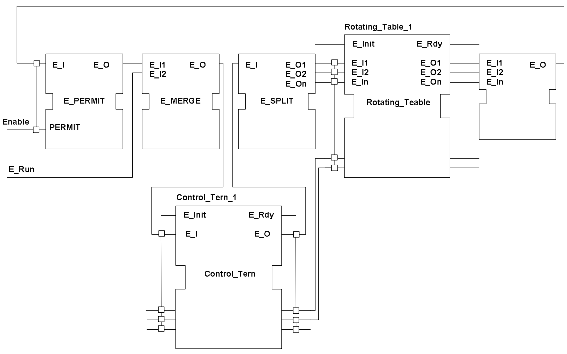


Fig. 6. FB sequence for the control system functional design.

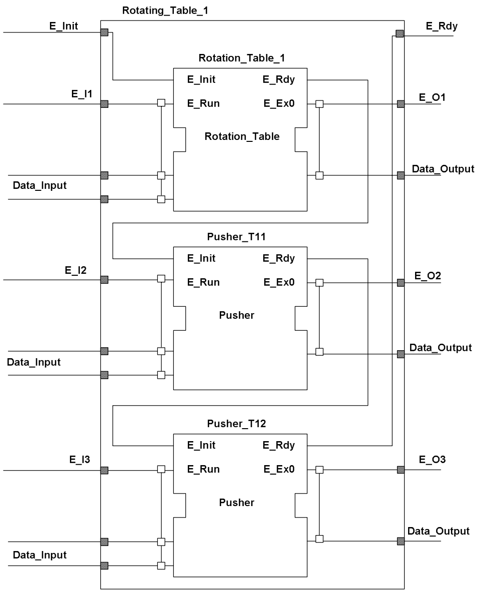


Fig. 7. Structure of the composite FB "Rotating_Table_1".

*4.4 Automation system implementation and simulation.*

Once the functional model is defined by means of the IEC 61499 standard, this can be tested through the bottom-up approach presented in (Carpanzano and Ballarino, 2002) before PLC code generation, so improving the reliability of the control software implemented on the real system, and reducing the overall development times and efforts significantly.
Now the implementation phase may be approached through the automation software development methodology proposed in (Carpanzano and Ballarino, 2002). Such a methodology follows a modular approach exploiting hierarchical SFC structures, see figure 8. In particular, complex SFC programs are developed so that the reachability, liveness and reversibility properties hold. Such programs are obtained by properly aggregating simpler SFCs for which the desired properties are satisfied.

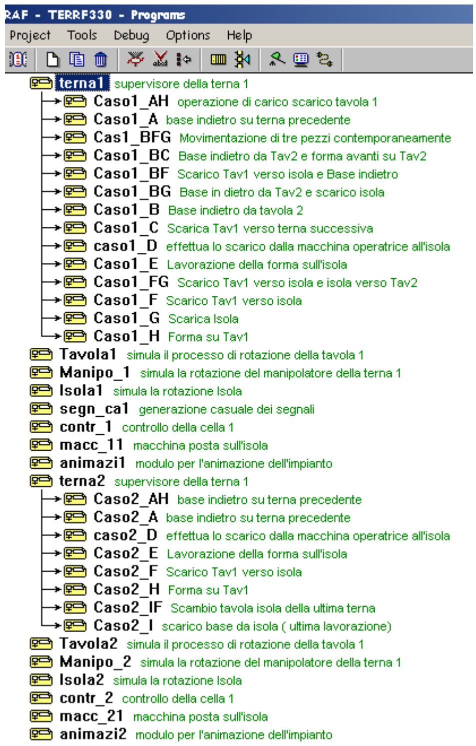


Fig. 8. SFC for the Tern.

In order to verify the functional correctness of the designed control solutions, closed loop discrete simulations have been performed in ISaGRAF. Therefore, proper simplified models of the Tables, Islands, Manipulators and Operation Stations have been represented in ISaGRAF by means of suitable SFC modules and data structures. A minimal input-output description of the controlled devices has been introduced, since this results to be sufficient for the functional testing of the designed control solutions.
To simplify the analysis of the simulation results a simple 2D graphic animation has been realized so as to verify the correctness and the consistency of the parts flows along the molecular transport line. Part of such a graphic animation is shown in Figure 9.

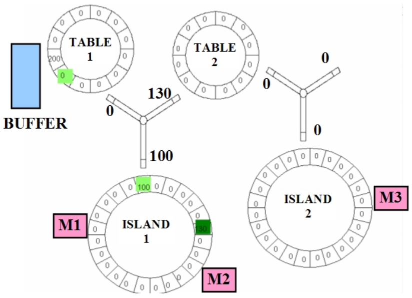


Fig. 9. Simulation and 2D animation in ISaGRAF.

*4.5 System integration and testing.*

Finally, the automation software can be downloaded on the target system. So, it can be verified that the desired requirements are fully met by the considered manufacturing cell.

In order to verify the software downloaded on the target system, before to integrate it with the real manufacturing plant, the automation system has been interfaced to a pilot plat which reproduce in scale 1:10 the molecular transport line (Donà, 2004), see figure 10. This technologic demonstrator is part of the ITIA Automatic Control Laboratory and it has been realized with the same number of input / output signals exchanged between the manufacturing plant and the automation system. This is because an automation system tested with the pilot plant can be directly connected to the real manufacturing plant without any other software verification.

The use of the considered methodology facilitates clear documentation and easy maintenance of the designed automation solutions, it also improves reusability and enhances fast integration of new features as well as easy reconfiguration of the designed automation system.

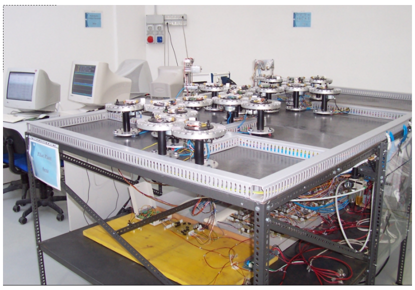

Fig. 10. ITIA Automatic control laboratory pilot plant.

## 5. CONCLUDING REMARKS

In the present work a development methodology for the design of manufacturing automation systems has been discussed, in particular the automation system functional specification has been described. In order to verify the control software behaviour a closed loop simulation has been performed. Moreover the test of the target system, configured with the designed control software has been carried out by connecting the automation control system to a pilot plant.

The definition of different control strategies for the adopted reference models will be subject of future work, as well as the application of the proposed framework to different manufacturing plants.

## REFERENCES

Bossi D., Carpanzano E., Cattaneo della Volta V. and Tilbury D. (2003). Modular Design of Logic Control Functions for RMS through Petri Nets and Sequential Function Charts. Proc. CIRP 2nd International Conference on Reconfigurable Manufacturing, Ann Arbor, Michigan, USA, paper A-22.

Carpanzano, E., and Ballarino, A. (2002). A structured approach to the design and simulation-based testing of factory automation systems . *Proceedings of the IEEE International Symposium on Industrial Electronics*, ISIE2002, July 8-11, L' Aquila, Italy.

Carpanzano E., Dallefrate D. and Jatta F. (2002). A Modular Framework for the Development of Self-reconfiguring Manufacturing Control Systems. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, IROS2002, Lausanne, Switzerland, September 30-October 4, Vol. 2, pp. 1960-1965.

Carpanzano E. and A. Cataldo. (2003). “Modular design and simulation based verification of the logic control code for an agile shoe manufacturing system”. Proc. Industrial Simulation Conference, ISC2003, EUROSIS, Valencia, Spain, pp. 290-294.

Donà M. (2004). Ideazione, progettazione e sviluppo di un dimostratore meccatronico per sistemi manifatturieri riconfigurabili.

Ferrarini, L. and Carpanzano, E. (2002). A Structured Methodology for the Design and Implementation of Control and Supervision Systems for robotic applications. *IEEE Transactions on Control Systems Technology*, **vol. 10, no. 2**, pp. 272-279.

Lewis R.W. (1998) Programming Industrial Control Systems using IEC 1131-3. Revised edition, The Institution of Electrical Engineers USA.

Lewis R.W.. (2001) Modelling Control Systems using IEC 61499. The Institution of Electrical Engineers USA.

Maffezzoni, C., Ferrarini, L., and Carpanzano, E. (1999). Object-Oriented Models for Advanced Automation Engineering. *Control Engineering Practice*, **vol. 7**, pp. 957-968.

Renè David. (1995) Grafcet: A Powerful Tool for Specification of Logic Controllers. IEEE Transactions on Control Systems Technology, Vol. N°3.

Schoop, R., and Strelzoff, A. (1996). Asynchronous and Synchronous Approaches for Programming Distributed Control Systems Based on Standards. *Control Engineering Practice*, **vol. 4, no. 6**, pp. 855-861.

Standard IEC 61131 (1993). Programmable Logic Controllers. IEC TC65/WG6.

Standard IEC 61499 (2001). Function Blocks for Industrial-Process Measurements and Control Systems. IEC TC65/WG6.