Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Robust filtering of PLC program for automated systems of production

Benlorhfar R., Annebicque D., Gellot F., Riera B.

CReSTIC, UFR des Sciences Exactes et Naturelles, Université de Reims Champagne-Ardenne
Moulin de la Housse - BP 1039, 51687 REIMS Cedex 2 – France
{rachid.benlorhfar,david.annebicque, francois.gellot, bernard.riera}@univ-reims.fr

Abstract: The proposed methodology enables to secure the control program of automated production systems. The principle is to place a robust filter inside the PLC of the automated production system to inhibit or authorize actions in the event of detection of errors in the control program. The proposed methodology rests on the one hand, on a rigorous analysis of the automated production system in order to formally build the set of constraints which constitute the filter using temporal and boolean logic. And on the other hand, a phase of modeling by finite-state automata which is necessary to validate the sufficiency of the constraints and to check the robustness of the filter as well as the safety properties using Model-Checking tools.

Keywords: Manufacturing systems, Automata theory, Programmable logic controllers, Robustness, Safety analysis, Dynamic modeling, Error control, Formal methods

1. INTRODUCTION

In parallel with the complexity increase of the automated production systems in term of quantity, requirements in communication, diversity of the components, etc., the requirements of users concerning the dependability (Villemeur, 1988) and the design assistance of control programs also increased. Indeed, the Programmable Logic Controller (PLC) constitutes the main establishment architecture of manufacturing system control, and is programmed with standardized languages (IEC 61131-3). But, the automatic control engineer does not have, until now, any real help for the control program design. In fact, only his/her competences and his/her experiment allow him/her to elaborating control programs. He/she is the guarantor that his/her programs allow to obtain safe functioning of the system. To ensure, in a formal way, the safety of the automated production systems is thus a real scientific challenge carrying important industrial issues. To guarantee that the control program of these systems is safe functioning is a major preoccupation of the systems owners. They are thus in need of methods and tools which allow to check that the control program suggested satisfying all the safety requirements.

The research works present in this paper concerns these problems. They talk only about automated production systems controlled by PLC, with only logical inputs and outputs. A methodology to design a robust filter for control program of automated production system is proposed. This robust filter composes of a set of safety and functional constraints. These constraints are formally defined from a structural and dysfunctional analysis of automated production system. They permit to ensure the dependability of the system and its products in the event of errors in the control program (Marangé et al., 2010). This methodology decomposes in two main steps: off-line approach and on-line approach. The off-line approach allows obtaining the constraints of the filter from a set of informal specifications. Then, a model-checking tool is used for checking and validation of these constraints (Schnoebelen et al., 1999). The on-line approach consists in implementing the robust filter into the PLC which controls the automated production system in order to ensure safety whatever the control program implemented in the PLC.

A modular approach of modeling is used in order to represent symbolically interactions between control part (program which implemented in a PLC) and an operative part (automated production system) (Balemi, 1992). Modeling is carried out by using finite-state automata (Rohee et al., 2008; Gouyon et al., 2004; Chandra, 2002). The safety properties are checked by a model-checker in order to validate the sufficiency of the constraints and to check the robustness of the filter. This methodology is the subject of an application. A virtual products sorting system coming from the simulation software of automated production system ITS PLC[1] is presented to illustrate this methodology.

2. METHODOLOGY OF CONSTRAINTS DEFINITION

An expert has to carry out the off-line approach. The methodology of constraints definition is carried out in several steps. The first step consists in a structural analysis of the automated production system in order to identify all the interactions between elements of this system. The second step consists in a dysfunctional analysis in order to identify all the hazardous situations. These hazardous situations are represented by execution sequences that the automated production system should not execute to avoid material degradations and products deteriorations. The last step allows

[1] Automated production systems simulator, freely downloadable in demonstration version at the following website: www.realgames.pt

to formally deducing a set of safety and functional constraints from the previous analyzes in order to build the filter of control program.

2.1 Structural analysis

The structural analysis allows obtaining a hierarchical decomposition of the system in materials and functional elements (Marangé et al., 2007). It is on this level that the functions present in the system are determined.

To determine the constraints of an automated production system, the expert must analyze the structure of the system in order to divide it into physical components. Each physical component is associated with a function of low-level hierarchical of the previous structural analysis of automated production system. Then, the expert must identify all the interactions between the components of the system, and these components and products (figure 1). These interactions represent the interconnections between the functions of the structural analysis. They can be due to a physical dependence of the components, a functional dependence between the components, or to dependence by the product. Indeed, the product evolution between the physical components of the system can generate dysfunctions. The product can occupy some dangerous positions, or prevent some evolutions of the system.

The system decomposition in elementary parts facilitates the modeling step of its physical components. Moreover, the interactions identification allows avoiding the combinatory explosions during the checking phase by model-checking tools (because the filter robustness is checked in a modular way). The interactions are analysed independently.

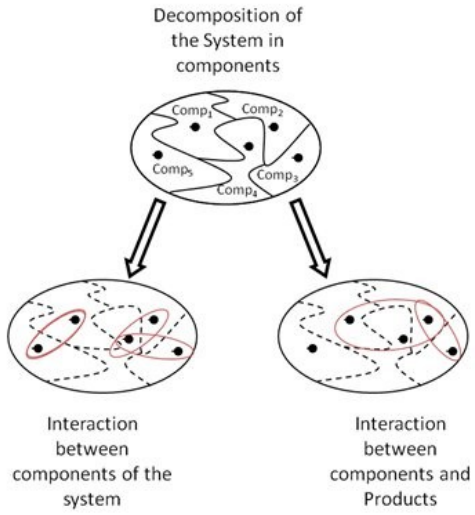

Fig. 1. Structural analysis of the system

2.2 Dysfunctional analysis

In order to determine the wrong functioning of the system, it is usual to proceed to a dysfunctional analysis (Villemeur, 1988). This analysis is specific to the dependability of the systems. This analysis makes possible to identify all the hazardous situations which a system should not reach. This method is generally based on an inductive reasoning making it possible to identify the causes and the consequences of a hazardous situation.

In the context of this methodology, and in order to build safety and functional constraints, the expert must proceed to a dysfunctional analysis of all the interactions of the structural analysis made previously. Each interaction is analyzed separately to determine all its hazardous situations (figure 2). The system should not be in one of these situations in order to not undergo material degradation or to cause deteriorations to the products. The expert must describe for each hazardous situation previously identified, their probable causes and consequences in order to facilitate the expression of safety and functional constraints.

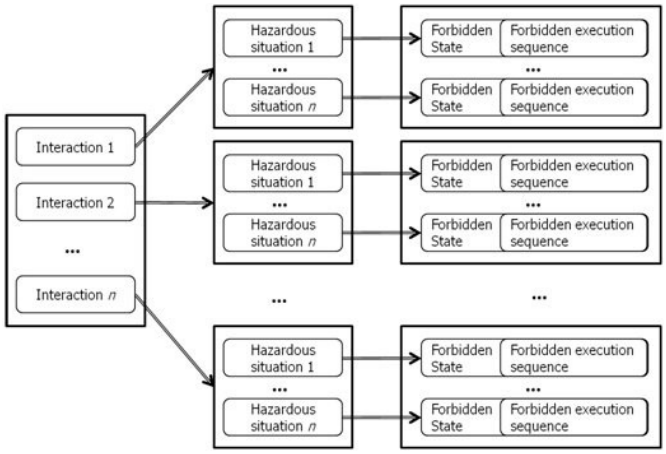

Fig. 2. Dysfunctional analysis of the interactions

The hazardous situation is represented by a given state of the system during the interaction. The hazardous situation is detected thanks to the sensors states and the products positions. The cause of a hazardous situation corresponds to a dangerous execution sequence. This sequence can involve degradations of the automated production system and products. This sequence represents errors in the control program which generate a bad manipulation of the system actuators. The consequence of a hazardous situation corresponds at the prohibited state that the system should not reach in order to preserve its integrity. The consequence represents the safety property which must be respected to ensure the dependability of the system.

This structural and dysfunctional analysis allows to the expert to have all the necessary information for the formal expression of safety and functional constraints which compose the robust filter of control program.

2.3 Definition of the safety constraints

In this methodology, the choice is made to formally enunciate the constraints by using logic CTL (Barragan, 2007). Indeed, the logic CTL facilitates the translation of specifications, in a literal language of the constraints to a Boolean language which is understandable and easily implementable in a PLC. The Boolean operators such as, the conjunction ($\wedge$), the disjunction ($\vee$), the implication ($\rightarrow$) or the negation ($\neg$) are used, as well as the temporal operator AG, (meaning "Always and Globally"). The constraints are formulated by using

logical implication. When a system is in a hazardous situation inducing a prohibited state, and that this situation is detected, then it is necessary to act on the control program of the system in order to avoid sending erroneous orders and causing damages to the automated production system and the products.

The detection of a hazardous situation during an interaction is realized through the sensors, information on the state of the system and information on position of the products. Indeed, it can be necessary to rebuild very important information to the expression of the constraints, when the information brought by the instrumentation of the system is not sufficient. The cases which require a information rebuilding are often due to an under-instrumentation of the physical components or to products positions which are not observable by the system sensors. However, it is always possible that information coming from the instrumentation is ambiguous. This information can refer to several states of the same physical component or to various products positions. In these cases, information rebuilding clarifying each state or position is necessary. The expert must combine information of the inputs and the outputs of the system in order to rebuild missing information. It is also possible to use temporal information. The expert can associate times envelopes with the inputs and outputs of the system to identify a particular state of a physical component or a specific product position.

The methodology suggested for the constraints definition is not formal. Even if their writing is generic, the manner of determining them depends on the system structure and technology of these physical components. This methodology also depends on expert competences and his/her experiment. Moreover, the filter of control program is implemented in PLC, of which sequential and cyclic functioning has necessarily consequences on the validity of the constraints like the simultaneous evolutions, or causality delays. It is thus necessary to check if the constraints are sufficient to guarantee the safety of automated production system and its products.

3. CHECKING OF THE FILTER ROBUSTNESS

3.1 Description of the methodology for constraints checking

In this section, a modular methodology for the constraints checking is proposed. The expert must develop models by the means of finite-state automata and for each interaction, all the functional chain (figure 3), in order to check if the safety properties are satisfied. The expert must realize the following tasks:

- Modeling the cyclic and sequential operating process of PLC.
- Modeling the most permissive control program possible of the physical components of the same interaction.
- Developing algorithms in order to rebuild missing information and modeling the safety and functional constraints.
- Modeling the nominal behavior of each physical component and defining the relations between these components of the same interaction to reproduce the dependences.
- Defining the relations between the product and the physical components of the same interaction where the product evolves in order to model the products flow.
- Making sure that the constraints are not too restrictive by checking some properties of liveness and reachability.
- Checking the safety properties in order to validate the constraints.

When the expert carried out the modeling of all the interactions of the system in the form of functional chains, he/she checks the safety properties by using model-checking techniques. If the safety properties are satisfied for each interaction, then the constraints are sufficient and the filter is robust to the control program errors; else the expert must redefine the safety constraints for the interactions which are not validated. He/she must cross-check, the safety properties in order to validate all the constraints and to be sure that the filter is robust.

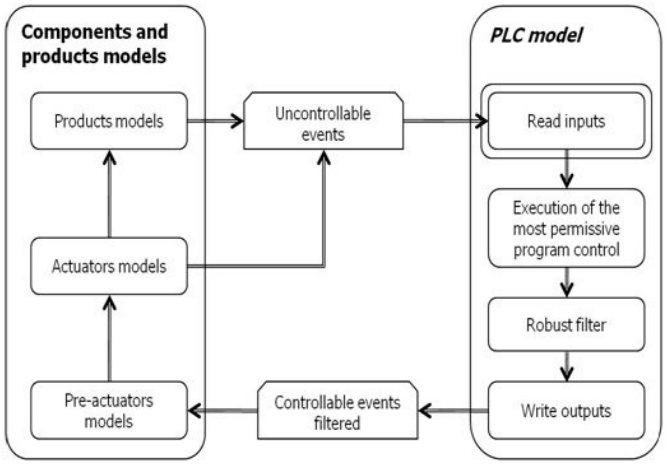

Fig. 3. Description of a functional chain

In the context of this methodology, the model-checker chosen is UPPAAL[2] (Behrmann et al., 2002). It allows from a definite initial state of the system components and products positions, to check the safety properties for all the possible ways of execution of the control program. If one of the properties is not checked then the constraints are insufficient. In this case, the expert must redefine the constraints by analyzing the traces of the model-checker, because UPPAAL is able to provide the execution sequence which engendered the prohibited state.

3.2 Modeling of the interactions

In order to reproduce the functional chain loop, an automaton schedules the execution cycle of the PLC model, of the physical components models and the products models (figure 4). In an initial state all the components are positioned in at-rest states and the products in their reference positions. When

[2] Integrated tool of modeling, validation and checking of real-time system. Freely downloadable at the following website: www.uppaal.com

the automaton evolves, it gives rhythm to the execution of the functional cycle. For each transition, a task is executed so that the environment of checking can evolve. It is supposed that all the changes of the sensors states are detected by the PLC model.

The scheduler automaton carries out sequentially and in a cyclic way the following tasks: The reading of the inputs consists in updating the PLC variables by recopying the sensors variables coming from the model of the automated production system. Then the control program is executed.

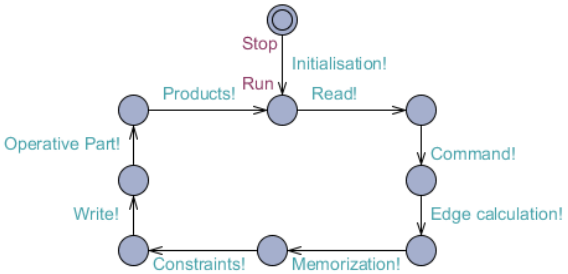

Fig. 4. Scheduling of the execution of the functional chain

Each component has its own control automaton. To generate all the possible sequences of the control signals of the components, a synchronized network of nondeterministic automata is used. For each execution cycle of the PLC, each automaton of control can evolve to another state or to remain in its actual state. The nondeterministic automaton is represented (figure 5) according to the number of controllable events (Σc) associated with the component that the model must to represent. It is sometimes possible, according to the components technology, to simplify the models by not taking into account the unrealizable states.

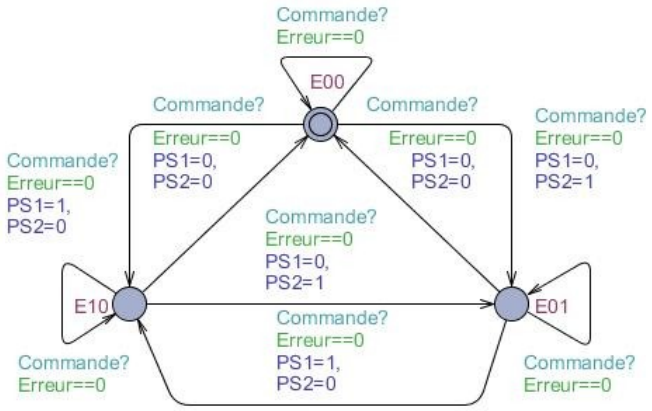

Fig. 5. Simplified model control of a component with 2 Σc

When the task of evolution of the control program is completed, a data processing is realized. It represents the memorizing step of the variables. This treatment allows calculating the rising edges and falling edges of the PLC variables. It also allows memorizing the states of the PLC variables for the next cycle. These calculations are used in order to detect the changes of states of the inputs and outputs of the PLC. The detection of the states changes is used mainly for the definition of the algorithms which are used to rebuild the necessary information for the constraints expression.

The constraints are represented by a synchronized automata network. When a constraint is violated a bit of error is activated. This bit is taken into account at the time of the update of the outputs. It allows inhibiting all the outputs in order to prevent all the possible system evolutions. If all the constraints are respected, the update is executed. The outputs of the PLC are recopied in the pre-actuators variables of the model components of the system. A function representing symbolically the commutation mode of the pre-actuators is associated with the update of the outputs. This function defines how the models of the actuators must evolve and must react to the possible changes of the control signals.

A function which represents the products evolution is also associated with the model of updating outputs. The products behavior during the interaction is represented symbolically by a set of rules. These rules constitute the mode of products displacement and their positioning during the interaction. The actuators states condition the evolution of these rules. A symbolic state variable is associated with each product to represent its position during the interaction. A sequence, representing the products flow during the interaction, generates the states of the system sensors. This sequence is represented by an automaton. The products models are thus a synchronized automata network reproducing the same sequence. Each product of the interaction evolves within its automaton according to its symbolic state variable.

The modeling approach of the actuators is the same as used for the control models. Then, they are associated with the system components according to the number of uncontrollable events (Σuc) which they must generate. These models reproduce symbolically the nominal behavior of each component. A discrete state variable represents the actuators position. A set of parameters of the physical structure of the component is taken into account to refine the model and to define the evolution rules of actuators of the system components.

This suggested methodology allows the checking of the filter robustness whatever the control program implemented in the PLC. The model checker examines all the possible executions of the control program and its environment in order to check the safety properties. The evolutions rules of the products and the components are symbolic. These rules reflect the nominal behavior of the system by taking into account a set of parameters. These parameters refer to products characteristics, structure and technology of the physical components of the system.

4. APPLICATION

4.1 Description of sorting system from ITS PLC

It is about a sorting system where the main goal is to transport cases from an entry conveyer to automatic elevators, sorting them by height (figure 6). This sorting system is entirely instrumented and composed of electric actuators and a set of photoelectric and inductive sensors.

The functioning principle of the sorting system is described as follows. The entry conveyer (A0) delivers randomly small and big cases; they are spaced by a safety distance. This

conveyer is equipped with a sensor (c0) which detects the crossing of the cases, disposed on pallets, towards the belt conveyer (A1). The size of the cases is detected at the entrance of belt conveyer. Two sensors are stacked. The sensor (c1) detects the passage of all the cases, whereas the sensor (c2) detects only the passage of the big cases. Then, the cases are transported by the belt conveyer towards a monostable turntable (A4). The crossing of the cases from the belt conveyer towards the turntable is detected by the sensor (c3). The sensors (c4) and (c5) detect the positions of loading and unloading of the turntable. The sensor (c6) detects the presence of the cases on the end of the turntable when it is in position of loading. Rollers with two directions of rotation (A2) and (A3) make it possible to switch the cases towards the delivery conveyers (A5) and (A6). Each delivery conveyer is equipped with a sensor to detect the crossing of the cases of the turntable towards the conveyer of exit (c7) on the right and (c8) on the left, and with a sensor to detect the crossing of the cases of the conveyer to the automatic elevators (c9) on the right and (c10) on the left.

The system includes some characteristics, the time cycle of the automatic elevators is largely lower than the time of shunting of the cases. It is supposed that once the cases are discharged on the delivery elevators, they leave the system. The activation of the outputs A2 and A3 simultaneously does not have any effect on the rollers.

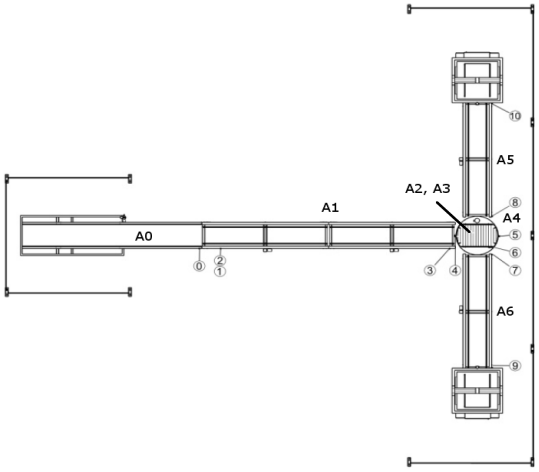

Fig. 6. Structure of sorting system

4.2 Identification of the interactions

The system decomposition in elementary components is realized by identifying all the elementary functions of the sorting system. Then, all the physical components of the system are identified (Table 1).

All the interconnections between the functions of the sorting system are realized by the evolution of the product on the system. The identified interactions are described below:

- I1: Presence of one or two case(s) between the entry conveyer and the belt conveyer.
- I2: Presence of a case between the belt conveyer and the turntable.
- I3: Presence of a case on the rollers of the turntable.
- I4: Presence of a case between the turntable and the right delivery conveyer.
- I5: Presence of a case between the turntable and the left delivery conveyer.

Table 1. System decomposition in components

Components	*Actuators*	*sensors*
Entry conveyer	A0	c0
Belt conveyer	A1	c1, c2, c3
Turntable	A4	c4, c5
Rollers	A2, A3	c6
Delivery conveyer at the left	A5	c8, c10
Delivery conveyer at the right	A6	c7, c9

4.3 Safety constraints

Before expressing the safety constraints of the filter of control program of the sorting system, it is necessary to determine the requirements in information on the cases positions during the interactions. Algorithms of rebuilding are defined to mitigate this lack of information:

For the I1 interaction, the P01 variable represents the number of cases present between the entry conveyer and the belt conveyer. The sensors c0 and c1 provide ambiguous information. A counter/discounter is associated with rising edge of sensor c0 and falling edge of sensor c1 to count the number of cases on the junction. For the I2 interaction, the P36 variable represents the presence of a case between the sensor c3 and the sensor c6. For the I4 interaction, the P67 variable represents the presence of a case between the sensor c6 and the sensor c7.

When the algorithms of rebuilding information are defined, a dysfunctional analysis is realized for each interaction. This analysis allows determining two prohibited states which should not be reached during the interactions. The first state corresponds to collisions between the cases at the time of their crossing on the junctions. And the second state corresponds to cases deteriorations during the steps of loading and unloading on the turntable. To prevent reachability of these states, constraints are expressed in the form of logical implication from literal expressions; these expressions correspond to execution scenes of the system known like prohibited. The constraints extracted from the analysis of the sorting system are given in the following.

For the interaction I1:

- If a case is detected between the two conveyers, then the entry conveyer must never be activated if the belt conveyer is not activated.

$$c0 \wedge c1 \wedge P_{01}\text{-}1 \rightarrow AG \neg (\neg A1 \wedge A0) \quad (1)$$

- If two cases are detected between the two conveyers, then the entry conveyer must never be activated.

$$c0 \wedge c1 \wedge P_{01}\text{-}2 \rightarrow AG \neg (A0) \quad (2)$$

For the interaction I2:

- If a case is detected at the end of the belt conveyer and the turntable is in loading position, then the belt conveyer must never be activated if the rollers (A2) are not activated.

$$c3 \wedge c4 \rightarrow AG \neg (\neg A2 \wedge A1) \quad (3)$$

- If a case is detected at the end of the belt conveyer, and the turntable is not in loading position, or another case is already present on the turntable, then the belt conveyer must never be activated.

$$c3 \wedge (\neg c4 \vee c6) \rightarrow AG \neg (A1) \quad (4)$$

- If the turntable is in loading position and no case is already above or if a case is loading, then the turntable must never be activated (rotation forbidden).

$$c4 \wedge (\neg c6 \vee P_{36}) \rightarrow AG \neg (A4) \quad (5)$$

For the interaction I3:

- If a case is detected on the turntable and if the turntable is in loading position, then the rollers (A2) must be never activated.

$$c4 \wedge c6 \rightarrow AG \neg (A2) \quad (6)$$

- If the turntable is in loading position, then the rollers (A3) must never be activated.

$$c4 \rightarrow AG \neg (A3) \quad (7)$$

- If the turntable is not in loading position and not in unloading position, then the rollers (A2) and (A3) must always be deactivated.

$$\neg c4 \wedge \neg c5 \rightarrow AG (\neg A2 \wedge \neg A3) \quad (8)$$

For the interactions I4 and I5:

- If a case is detected on the turntable in unload position or if the case is unloading on the right or on the left delivery conveyer, then the turntable must always be activated.

$$(c5 \wedge c6) \vee P_{67} \vee c8 \rightarrow AG (A4) \quad (9)$$

The step of constraints definition was described exhaustively, whereas the steps of modeling and checking were not described. But, methodology was completely applied to all the interactions of the sorting system and the sufficiency of the constraints was checked and validated. The logical constraints and the algorithms to rebuild information were implemented in a real PLC and several scenarios of control programs were tested to control the sorting system in full safety.

5. CONCLUSIONS

The suggested methodology in this paper is original. It allows ensuring the reliability and the dependability of automated production system by using robust filter of the errors of control program. Moreover this methodology is completely independent of the functional specifications which can change during the lifecycle of automated production system. Libraries of models and application are under development to help an expert to realize the off-line approach and several ideas in order to diversify the on-line use of the filters are in the course of experimentation.

Acknowledgment

The authors want to thank the Champagne-Ardenne region for their financial support under the project CPER-MOSYP (French acronym for Performance Measurement and Optimization of Production Systems).

REFERENCES

Balemi S., (1992). *Control of discrete event processes: theory and application*, PhD thesis, Swiss Federal Institute of Technology, Zurich, Suisse.

Barragan Santiago I., (2007). *Élaboration de propriétés formelles de contrôleurs logiques à partir d'analyse prévisionnelle par arbres de défaillances*, PhD thesis, École Normale Supérieur, Cachan, France.

Behrmann G., Bengtsson J., David A., Larsen K.G., Pettersson P., Yi W., (2002). Uppaal implementation secrets, In Proc. of *7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems*, Oldenburg, Germany.

Chandra V., Kumar R., (2002), *A event occurrence rules based compact modeling formalism for a class of discrete event systems*, Mathematical and computer modeling of dynamical Systems, Vol. 8, N°1, pp 49-73.

Gouyon D., Pétin J.F., Gouin A., (2004). *A pragmatic approach for modular control synthesis and implementation*, International Journal of Production Research, vol. 42, n° 14, pp. 2839-2858, Taylor & Francis Publisher.

International Electrotechnical Commission, (1993), PLCs – Part 3: programming languages, Publication 611131-3.

Marangé P., Gellot F., Riera B., (2007). Remote control of automation systems for D.E.S. course, *IEEE Transactions on Industrial Electronics*, 54(6):3103-3111.

Marangé P., Benlorhfar R., Gellot F., Riera B.,(2010). Prevention of human control errors by robust filter for manufacturing system, *In Proc. of The 11th Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, Valenciennes, France.

Rohée B., Carré-Ménétrier V., Riera B., (2008), Proposition of completeness property to perform the plant modeling for manufacturing applications, *In Proc. of The 17th World Congress International Federation of Automatic Control*, Seoul, Korea

Schnoebelen P., Bérard B., Bidoit M., Laroussinie F., Petit A., (1999), *Vérification de logiciels : Techniques et outils du model-checking*, Vuibert, France

Villemeur A., (1988), *Sûreté de fonctionnement des systèmes industriels*, Eyrolles, France