



HAL
open science

REMOTE CONTROL OF AUTOMATION SYSTEMS FOR TRAINING

Pascale Marangé, François Gellot, Bernard Riera

► **To cite this version:**

Pascale Marangé, François Gellot, Bernard Riera. REMOTE CONTROL OF AUTOMATION SYSTEMS FOR TRAINING. IFAC/IFIP/IFORS/IEA Symposium, Analysis, Design, and Evaluation of Human Machine Systems (ADEHMS'07), Sep 2007, Séoul, South Korea. pp.CD. ⟨hal-00385512⟩

HAL Id: hal-00385512

<https://hal.science/hal-00385512v1>

Submitted on 19 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

REMOTE CONTROL OF AUTOMATION SYSTEMS FOR TRAINING

Pascale MARANGE, François GELLOTT, Bernard RIERA

*CRéSTIC UFR des Sciences Exactes et Naturelles
Université de Reims Champagne-Ardenne, Moulin de la Housse, BP 1039, 51687 REIMS Cedex 2*

Abstract: We are interested in the remote use of automated systems by students within the framework of Discrete Event Systems teaching. One of the most important problems in this case, is to guarantee that the proposed PLC (Programmable Logic Controller) control is correct: in the other word, respecting the specifications given by the teacher. Hence, there is a validation problem of the control. To solve this problem, we propose an on-line validation approach based on a filter placed between the PLC and the plant. A second problem is the system dimension. Indeed, we propose an approach to keep a global vision adapted at the level of learners. For that, we propose a functional identification. We applied this method to an original project where some 9 year old children discover automation by programming a tablets packaging system.

Keywords: Control Validation, Discrete Event Systems, Functional Identification, Learning

1. INTRODUCTION

We propose in this paper, to introduce a method to validate logic controller programs adapted to the teaching of DES (Discrete Event Systems) (Wonham, 2004) and the different controller programming languages described in the international standard IEC 1131-3 (IEC, 2002). This standard specifies the syntax, semantics and display for the following PLC programming languages: Ladder Diagram (LD), Sequential Function Charts (SFC), Function Block Diagram (FBD), Structured Text (ST), and Instruction List (IL). After theory, students have to practice on real PLC and real or pedagogical platforms (called also systems or plants), which allows student, to test and to see the real effects of his PLC program. The use of real systems for teaching raises three problems. The first problem concerns the safety of users (students and teachers) and materials. Indeed, an error on the controller program can involve a material breakdowns and/or the setting in danger of the users. The second problem is the necessity to be able to detect and bring an explanation to possible errors made by students. The thirds problem is adapted the system with the student's level. In this paper, we propose to give some answers to these three problems. To keep a global system vision, we propose to make a functional identification to adapt dimension at the learner level. We will see this definition in section 2. However, that involves a problem of controller validation that will be described in section 3. We propose a validation approach based on a filter designed by the mean of the definitions of safety constraints (what the system should not do) and of liveness constraints (what the system must do). These constraints are described in

the section 4. In order to test our approach, we propose 9year-old children to program a real system on a PLC. This project is presented section 5. Lastly, we will finish by the conclusion and some futures works.

2. DEFINITION OF ACTIONS POSSIBILITIES

It is obvious that the definition for a control problem must be adapted to the student. The level, the knowledge and the competence of analysis required are not the same for a student who discovers the automatism or for a student who follows a specialization course. To give training interesting, it is important to place the real systems at the disposal of learners. However, that induces necessarily some constraints which influence considerably the difficulty level of a control problem. We formalize in the following paragraphs the parameters which influence directly the difficulty degree of a control problem. Voluntarily, the "student point of view" and his perception of the difficulty level will not be considered.

2.1 Parameters related to the difficulty

The difficulty concept is to be connected to the complexity concept which is characterized by the high number of variables, by the interaction between its variables... The perception of the system complexity, its analysis, and its model are specific to the teaching objectives. Work of Lind (Lind, 1994) considers that the production systems can be defined according to two axes: "Means-Ends" and "Whole-Part". According to the author, by distinguishing the "Means" from the "Ends", a system can be described in terms of "Ends", "Functions" and "Physical

Components”. Moreover, each one of these descriptions can be given at various levels of the “Whole-Part” axis decomposition.

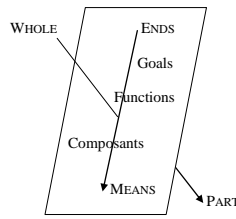


Fig. 1. “Means-Ends” and “Whole-Part”

We can bind this vision of the system to the difficulty level of a control problem. A plant (PO) is made up and is perceived by a fixed set of sensors and actuators and not easily modifiable. The simplest solution to simplify the system and its control is to consider certain PO elements. However this solution does not make it possible to preserve the global system vision. The approach that we propose consists to perceiving the system according to various levels of the axis “Means- Ends” in control to simplify the system perception. It is obvious that the complexity of the control is not directly related on the plant dimension but to the running specifications. According to our teacher experiment, the difficulty of a control problem can be expressed through 3 parameters, dependent for each other, and characterizing the control which answers the running specifications: dimension, hierarchization and synchronization between the elements.

2.1.1 Dimension

The control dimension is directly related to the number of subsets having to be controlled. It depends on the low decomposition level on the sensors and actuators number necessary to design the control described by the running specification. The granularity is high-quality, more the analysis effort is important for student and more the difficulty level is higher. It seems to us that if student is expert, he is able to work on the low level of the axis “Means-Ends”, i.e.: on the sensors-actuators level.

2.1.2 Hierarchization

The management of a “normal” cycle without taking account of the various operation modes requires a simple sequential control. On the other hand, a “complete” running specification integrating various operation modes and emergency stop requires a solution of hierarchical control. The specification stage is then more difficult. In practice, hierarchization appears through the forcing instructions of Grafcet for example.

2.1.3 Synchronisation

The two preceding parameters will reveal synchronizations due to the definition of the system dimension (simultaneous evolution of events) and to the hierarchization generated some priorities between the various operating modes. However, the control problems can also require some particular structures

of events management or common space.

2.2 Methodology to adapt the difficulty level

The definition of running specifications along the axis “Means-Ends” will make it possible to modulate the difficulty induced by each of the 3 previous parameters, keeping a global vision of PO to be controlled. We thus propose to adapt the difficulty level by establishing a functional identification of the system and by giving more or less autonomy to student. The functional identification goes makes it possible to define functions characterizing an instruction or a set of instruction more or less complex.

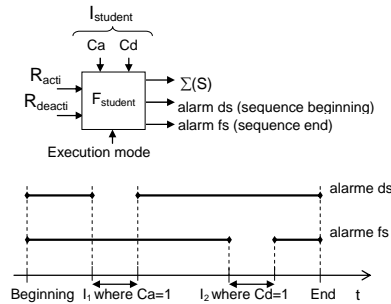


Fig. 2. Function definition

The functional identification made by the teacher thus will define the actions possibility of student i.e. the functions ($F_{student}$) which he has at his disposal to control the system, as well as the activation conditions Ca and the goals to reached represented by the deactivation conditions Cd (figure 2). It makes it possible the teacher to encapsulate concepts of synchronization and structures not adapted to student inside a function. According to the learners’ level, the teacher can choose to carry out the function in autonomous mode or not. In the first case, the activation and the deactivation of the function will be made automatically when the activation and deactivation conditions are respectively true. In the second case, the student must activate or deactivate the function when the conditions are present, if not he/she is informed of an error. The principle of function execution is as follows. The activation or the deactivation of a function will be effective only if the conditions of activation or deactivation are present. If the student makes the request of function activation (R_{acti}) in the interval I_1 , where the activation conditions are present (Ca_i), the sending is correct and the function carries out the set of the “outputs system” ($\Sigma(S)$). On the other hand, if the request is made outside I_1 , an alarm ds_i is emitted. It is the same for the alarm fs_i which is emitted, if the request for deactivation (R_{deacti}) is not made in the interval I_2 . This approach of modelling by function makes it possible to highlight traditional control errors made by the students such as the bad activation or deactivation of an instruction.

3. VALIDATION APPROCH

Work in the field of the automatic control validation aims to assure that mathematical properties are respected by model (Emerson and al, 1990), (Canet,

2001), (Lampérière and al, 2000). The work undertaken within the framework of tool UPPAAL (Behramm and al, 2004) defines three types of properties: attainability, safety and liveness. We chose to use the safety constraints: what the system should not do, and liveness constraints: what the system must do compared to the running specification. The validation can be considered off-line or on-line. In the first case, the control is completely validated before being sent towards the operative part (Machado, 2006). Within this framework, we proposed an off-line approach (Tajer and al, 2006) based on the Ramadge-Wonham supervisory control theory (Wonham and Ramadge, 1987) and the synthesis algorithm of Kumar (Kumar, 1991). The suggested approach makes it possible to guarantee that the control behaviour is sure, deterministic and without dead-locks. However, it presents several disadvantages: the combinative explosion, the difficulty to introduce the product and to give a comprehensible explanation to learner. So, we directed our work towards an on-line approach of control validation.

By this validation approach, the idea is to inhibit the evolutions which can lead the system in a situation of deterioration, of setting in danger of the operators or which does not respect running specification. Cruette's work (Cruette, 1991) for the monitoring of the automated systems proposes to intercalate a filter between the plant and control. The filter ensures on the one hand coherence between the sent control and that expected, and on the other hand coherence between the evolution of the expected plant and that produced. This idea of an on-line approach by filter is taken up partially and adapted to ensure the control validation i.e. with each new control evolution, the filter receives in inputs: the outputs evolution (controllable events: actuators) coming from the control designed by the student as well as the evolutions of the plant inputs (uncontrollable events: sensors). From this information and a specifications set, the validation filter authorizes or not the sending of the control evolution to plant. If the instruction is validated by the filter, it is sent to the plant, if not two cases are possible. Firstly, the suggested control does not ensure the safety of the system and the users. In this case the system is stopped and the learner is advised of an error. Secondly, the control does not validate the specifications completely but is not dangerous. In this case the instruction is sent towards the system and the learner is informed of the non respect specification.

The controller program validation consists in making sure that this one corresponds to the running specifications (i.e. modelling intentions of the teacher). That requires validation of the liveness and safety properties (also called constraints) and response time. In this approach, we focus on two types of student's errors:

- Those that do not respect safety constraints (what the system should not do). In other words, PLC program involves critical effects for human beings or materials on the plant.

- Those that do not respect liveness constraints (what the system must do). In other words, PLC program does not respect running specifications.

At the stage of validation, two cases can then occur:

- No error has been detected i.e. the controller respects the specifications imposed on the system.
- If an error is detected, the system has to be stopped and student has to be informed on the error. We suppose that the student has to perform himself the necessary improvements on his control program.

In this paper, we suppose that student uses a PLC and PLC programming languages like Ladder diagram (LD), Sequential Function Charts (SFC), Function Block Diagram (FBD), Structured Text (ST), and Instruction List (IL). It is interesting to note that in Europe, the Grafset (IEC, 2002) is used as specification and programming language. Several works on e-learning use Grafset. For example, Diez and his colleagues (Diez et al., 2006) propose a course and interactive exercises for the Grafset training. Our approach is complementary because we propose to the student to perform his own PLC program. After, the PLC program will be validated and further information about errors will be given.

4. VALIDATION FILTER

To ensure the safety and the correct functioning of the system, the validation filter uses some specifications to detect and to bring an automatic explanation. The specifications definition is not easy. Thus, we propose to carry out a distinction on their role (safety or liveness) and on their intrinsic characteristics (combinational, sequential, dynamic or static).

4.1 Safety validation of system

The safety constraints characterize what the system should not do. It seems us important to place the safety constraints at the sensors - actuators level. Three types of constraint of safety are defined.

4.1.1 Static safety constraints (SSC)

The static safety constraints express physical and technical impossibilities of the system elements. The static safety constraints depend only on the controllable states (S_c). The syntax is: $S_{c_i} \wedge S_{c_j} = 0$.

4.1.2 Dynamic safety constraints (DSC)

The dynamic safety constraints relate to the occurrence of an event which is not compatible with a system state. The event corresponds either at the controllable event (E_c), or uncontrollable event E_{uc} :

- In the first case, the constraint is written in the following way: $S_{uc_i} \wedge E_{c_j} = 0$. Indeed, if the deactivation conditions (S_{uc}) are present, the sending of the associated controllable event is prohibited.

- In the second case, the constraint is written: $S_{c_j} \wedge E_{uc_i} = 0$. Indeed, as soon as the deactivation

conditions are occurred, the actuator must be deactivated (S_c).

The safety constraints make it possible to protect the system against deteriorations. For these constraints, the validation filter can be placed in the control part (PLC). The validation filter prohibits the sending of an instruction if this one does not respect the set of the safety constraints. In other words, the set of safety constraints authorizes the most permissive safety control.

The definition of the safety constraints is re-used at the functional level, in a redundant way to bring an automatic explanation to the errors made by learner:

- If functions cannot be activated at the same time: $F_i \wedge F_j = 0$

- If the deactivation condition of a function is present, the sending of the function is prohibited:

$$Cd_F_i \wedge \downarrow F_i = 0.$$

- If the activation condition is not present, the function cannot be activated: $\neg Ca_F_i \wedge \uparrow F_i = 0.$

- As soon as the deactivation condition is present, the function must be deactivated:

$$F_i \wedge \uparrow Cd_F_i = 0$$

It is necessary now to determine if functioning is correct compared to the running specification. For that it is proposed to set up liveness constraints.

4.2 Liveness validation of system

The control validation compared to functioning, goes through by the definition of liveness constraints (what the system must do compared to the running specification). Contrary to the safety constraints, the liveness constraints are placed only at the functional level. Two types of constraints are defined: combinational and sequential liveness constraints.

4.2.1 Combinational liveness constraints (CLC)

The combinational liveness constraints allow activation or deactivation when the conditions are present. The combinational liveness constraints are defined in a similar way to the dynamic safety constraints. For example the function F_i can occur only under the condition Ca_i : $\uparrow F_i \wedge Ca_i = 1$ or the function F_i must be deactivated when the condition Cd_i is true: $\downarrow F_i \wedge Cd_i = 1.$

4.2.2 Sequential liveness constraints (SLC)

By the sequential liveness constraints, the function sequencing is representing. To define the functions that had to be carried out before the activation of the other one. The idea is thus to represent the sequence described by the running specification without to describe one unique behaviour. The logical equations do not make it possible to manage this sequential aspect simply.

The possibility to carry out a function compared to the expected behaviour depends on the system situation, i.e.: the functions which have been carried out. We point out that the possibility to carry out a function compared to the system state is expressed

by the combinational liveness constraints. To take into account the functions sequencing, for each function, we define the deactivation conditions and the functions which had to be fulfilled. In the same way, the function execution will influence the future behaviours and thus the functions which will not be realizable any more. To express the functioning sequencing, it is proposed to use Allen's algebra (Allen, 1983). For each function, we define Grafset with the states {not carried out, in execution, carried out} (figure 3). Grafset evolves at the same time as the control. Grafset makes it possible to know the functions authorized or not compared to at functioning awaited.

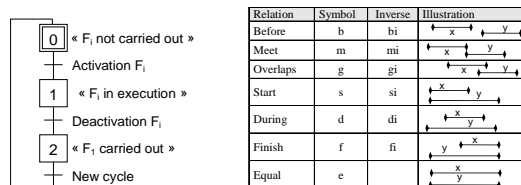


Fig. 3. Sequential liveness constraint

According to the functional identification, if the functions are carried out the ones after the others or in parallel, all the constraints will not be defined. Indeed, if the execution is in "non autonomous" mode, the function is deactivated automatically when the deactivation conditions are present. In this case, it is not necessary to define the dynamic safety constraints on the uncontrollable events.

5. APPLICATION

In order to test the proposed validation methodology, we applied it on an original operating industrial automation system named "Productis". We applied this method to an original project where it was proposed to 9 year-old children, to discover automation, by programming a tablets packaging system.

5.1 System presentation

The system used for this project is the machine "Productis". This system allows the packaging of tablets (figure 4). The system is composed of 5 stations and a conveyor:

- Manual loading of the pallet (bottle and stopper) (station 5),
- Product batching through tablet counting (station 1 and station 3),
- Bottle closing (station 2 and station 4),
- Bottle evacuation (station 4).

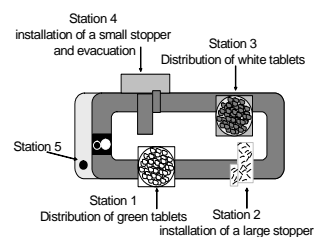


Fig.4. “Productis” Machine

In order to make the activity of programming funny for the child, we propose the following original scenario. The instructions to use the machine have been lost. So, it is impossible for us to manufacture tablet to heal sick fairies. Children have to find the running of the machine in order to manufacture specific tablets. We have to adapt the vocabulary used to describe the system, at the age of the child.

5.2 Training use

In order to test the proposed validation methodology, we applied it on an original operating industrial automation system named “Productis”. The idea is to enable people who start to study the Discrete Event Systems field to program a complete sequence of the “Productis” process and to test it remotely in a full safety. For the moment, the validation filter includes static safety, (SSC) dynamic safety (DSC) and combinatory liveness (CLC) constraints. The specification consists of processing one and only one

bottle. The constraints implementation is done directly in PLC by using the Ladder language. The training proceeds in the following way (figure 5):

- Control engineer student proposes a sequence for the control via a dedicated HMI (Human-Machine Interface). Each step of the sequence is sent one after the other to PLC.
- During the execution of PLC program, the outputs are validated by the filter. The PLC program is structured in two parts as follows :
 - Treatment of the main program of control
 - Then by passage in the filter: treatment of the safety and liveness constraints thus allowing us to detect errors. Indeed, according to the evolution of the control compared to the new inputs, if one of the constraints is not respected, we prevent the system evolution,
 - Update of the outputs validated by the filter.
- Results from validation filter are sent by the PLC to the process and to the HMI:

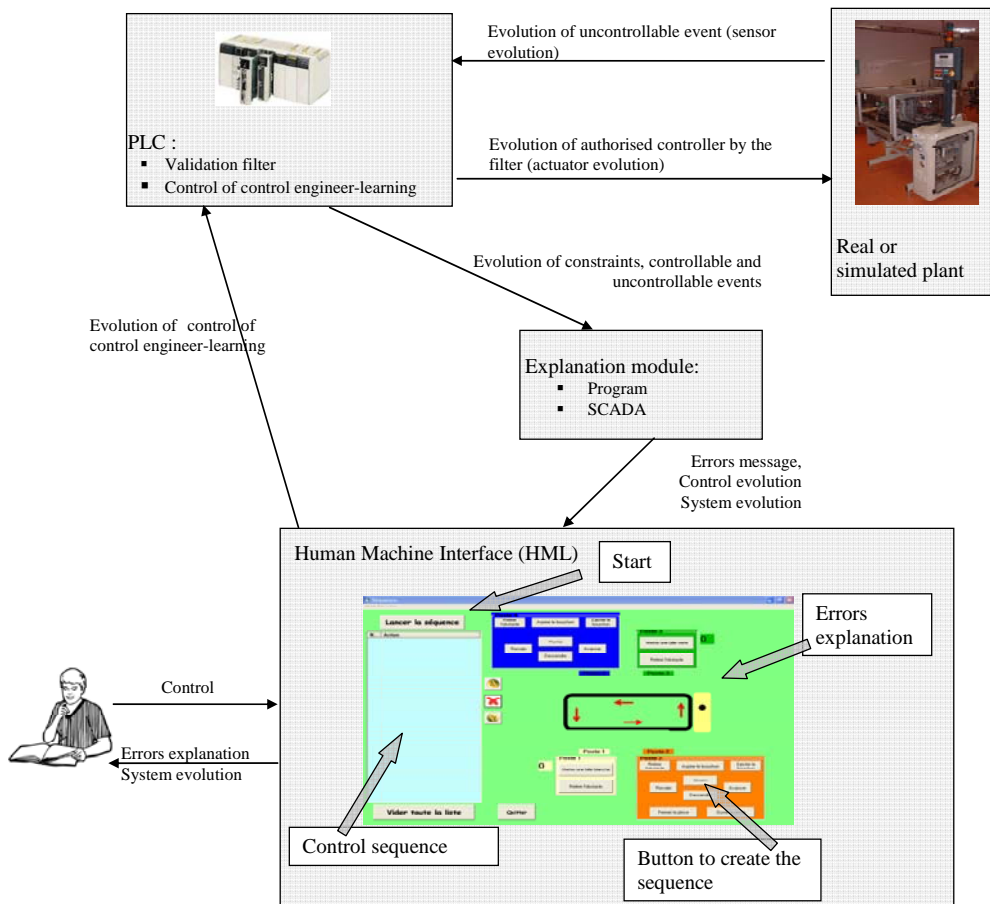


Fig. 5. Training approach

- If the filter validates all constraints, outputs are authorized and the sequence on the HMI can follow up.
- If the control does not validate the set of the constraints, outputs are not sent to the “Productis” system and the explanation module gives explanation associated with constraints which are not validated



a) “Step by step” mode b) Sequence mode
Fig. 6. Human Machine Interface

The idea was to collaborate with a teacher of primary school, in order to young children discovered the automatism field. During a previous project with

children (Riera et al., 2005), they have to pilot in a “non autonomous” mode the system through HMI and SCADA software. This time, we wanted to allow the child to discover and control really the system to programming his own sequence. For that, we propose them two HMI. In the first, the child has at his disposal an HMI (figure 6.a) with 20 command buttons. The 20 buttons represent the 20 functions of the Productis. In this activity, the child has to understand the function behind each button. During the second activity (figure 6.b), the child programs his own sequence of functions to bottle medicine tablets. The sequence execution is validated on-line. When the safety constraints are respected, sequence runs normally. If a safety constraint is violated, the child is informed with an explanatory alarm and the Productis is stopped and returns to its initial position.

6. CONCLUSION

This paper presents a proposal of a controller validation adapted to the DES training. We have showed that practical training is necessary on real or simulated plant but this use raises some problems. A validation module seems interesting to detect and explain an error. In this paper, we proposed an approach based on the definition of safety and liveness constraints. In a first application, we have only considered SSC, DSC and CLC. We are interesting at a second problem: the use of complex industrial systems within the framework of teaching where the system dimension must be adapted to the level of learners. The ideas have been tested on a real application with 9-year-old children. We raised the validation problem of SCL; we should define a methodology to define the running specification and to help the teacher to define the set of constraints.

REFERENCES

- Allen J. F., “Maintaining knowledge about temporal intervals”. *Commun. ACM*, 26(11):832–843, 1983.
- Behramm G., David A., Larsen K.G., A tutorial on UPPAAL, novembre 2004
- Canet G., Vérification automatique de programmes écrits dans les langages IL et ST de la norme IEC, *thèse de doctorat*, Ecole Normale Supérieure de Cachan. Décembre 2001
- Cruette D., Méthodologie de conception des systèmes complexes a événements discrets : application à la conception et à la validation hiérarchisée de la commande de cellules flexibles de production dans l’industrie manufacturière, *Thèse de doctorat, Université de Lille*, 1991
- Diez J.L., Valera A., Navarro J.L. Vallés M., Encinas A., “An interactive course on “logic controllers design using Grafcet””, *Proceeding of 7th IFAC symposium on Advances in control Education, ACE’06*, Paper WeC02.1sur le CD-ROM, Madrid, Spain, June 21-23, 2006
- Emerson E.A., Van Leeuwen D.J. Temporal and modal logic, *editor Handbook of the theoretical Computer Sciences*, vol. 9, chapitre 16, pages 995-1072, 1990
- International Electrotechnical Commission, “Preparation of function charts for control systems, International Standard”, *CEI/IEC 60848*, février 2002.
- Kumar R., Supervisory Synthesis Techniques for Discrete Event Dynamical Systems, *Thesis for Ph. D. Degree*, Université du Texas, 1991.
- Lampérière S., Lesage J.J, Formal verification of the sequential part of PLC programs, *Proc. Of 5th IFAC Wodes*, pp 247-254, Ghent, Belgium, August 2000
- Lind M., Modeling Goals and Functions of Complex Industrial Plant. Applied Artificial Intelligence, Vol8 No.2, April-June 1994
- Machado, Influence de la prise en compte d’un modèle de processus en vérification formelle des systèmes à événements discrets, *Thèse de doctorat de l’école normale supérieure de Cachan et de l’université de Minho*, juin 2006
- Riera B., Gellot F., Marangé P., Chemla J-P., Sayed Mouchawed M., “Un projet original en commande et supervision des systèmes automatisés : Des enfants de 5ans au secours d’animaux malades ! ”, *CETISIS’05*, Nancy, France, 25-27 octobre 2005
- Tajer A., Marangé P., Gellot F., Carré Ménétrier V., “Synthèse d’une commande supervisée à base de contraintes logiques“ *revue électronique e-STA* , 2006
- Wonham W.M., “Supervisory control of discrete-event systems”, *ECE 1636F/1637S*, 2004.
- Wonham W. M., Ramadge P.J., On the supremal controllable sublanguage of has given language, *SIAM J Control Optimization*, flight 25, n°3, p.637-659, 1987