REMOTE CONTROL OF AUTOMATION SYSTEMS FOR D.E.S COURSES

Pascale Marangé, François Gellot, Bernard Riera

REMOTE CONTROL OF AUTOMATION SYSTEMS FOR D.E.S COURSES

Pascale Marangé, François Gellot, Bernard Riera

***Abstract*—Objectives of technical courses are knowledge and know-how transfer to students. In the case of Discrete Events Systems courses, it is important for learner to control locally or remotely real systems (or plants) composed of many sensors and actuators. The use of these devices poses several problems. Firstly, it is difficult to adapt them to the student's level (from beginner to expert). Secondly, these systems are generally designed with industrial components. An error on the control-command design can involve safety problems and breakdowns.**

In this paper, they propose an original solution to solve these 2 problems. In order to guarantee the safety of the operators and the equipment, an approach using a validation filter is proposed. It is based on the definition of logical constraints which, should in no case be violated. In order to adapt the difficulty level, it is proposed to modify the level of automation. For that, the functional dimension of the automation system is modified to adapt the student's level of autonomy. The level of automation is defined by the teacher by the mean of a functional analysis of the system. In order to validate the approach, they applied it to an original project with 10 year old children on a packaging system. The aim of the project was to enable "young novice control engineers" to perform their first Programmable Logic Controller program to control the whole system.

***Index Terms*— Discrete events system, validation, control, functional identification.**

I. INTRODUCTION

Nowadays, the use of Communication and Information Technologies is a reality in the automation field. Indeed, one can find a massive use of the Ethernet network, as well as the level of the inputs/outputs (sensors and actuators), as in the communication between Programmable Logic Controllers (PLC). The use of TCP-IP, Web server in the PLC is to send Email or to connect to data bases like Oracle, Sql server or My SQL are classical applications. Thus, remote access to the controller via Internet has become a reality, allowing for example PLC programming, supervisory control and plant maintenance and tele-operation [1].

Internet provides different possibilities for the "practical" teaching of Discrete Events System (D.E.S) theory. The idea that they have developed [2] is to give the possibility to students to use, in a remote way, some professional materials (controller and plant) and software packages.

In the field of the automatic control of continuous processes, use of virtual and/or remote laboratories for teaching is well known. They can quote for example, Metzger's work [3] which uses Internet to reach virtual control devices for the teaching of distributed control devices. Remote use of real systems in feed-back control can be found in relevant literature [4] [5]. Remote Virtual labs can be very useful to illustrate advanced control applied to classical systems like the inverted pendulum [6].

On the other hand, only few papers concern the D.E.S teaching and the use of real or simulated control/command systems (controller) and manufacturing systems (plant) in a local or remote way. Hassapis [7] proposes to use simulators of DCS (distributed computer systems) and PLC integrated in an interactive electronic book. Bellmunt's work [8] aims at making the laboratory platforms available through the Internet in order to allow the use of professional practices in e-learning-based courses. However, these approaches do not consider the problem of system safety and the way to adapt the use of real system to the student's level. Indeed, these systems are generally designed with industrial components. A control-command error in the design can involve safety problems and breakdowns. Technologies today allow a remote use of plant. That makes it possible to improve the availability of the work practice rooms but ask pedagogy and safety questions.

In this paper, they propose an original solution to solve these 2 problems. In order to guarantee the safety of the operators and the equipment, an approach using a validation filter is proposed. It is based on the definition of logical constraints which should in no case be violated. In order to adapt the difficulty level, it is proposed on the one hand to modify the level of automation. On the other hand, it is proposed to adapt the student's level of autonomy. The level of automation is defined by the teacher by the mean of a functional analysis of the control-command specifications.

The first part of the paper deals with the specificities of the D.E.S teaching, which depend on the concerned public (from novice to expert) and the objectives (discovery, initiation, specialization). D.E.S teaching activities can use Internet at different levels, from discovery to specialization, to supply to students, PLC connected to real or simulated plant. They focus on the problem of the controller design where students start from specification given by the teacher to propose an implementation into a PLC of a solution to control a "real" operating industrial automation system.

One of main difficulty is to adapt the plant system to the different users, keeping the device as a whole. They define in the paper the difficulty level of a Logic Controller Design (LCD) by means of 3 parameters: dimension, synchronization and hierarchization. In order to adapt the difficulty level to learner without withdrawing the global plant vision, the approach presented is based on the modification of the system level of automation. For that, they propose to modify the functional dimension of the plant and the student's level of autonomy.

Remote use of real plants causes problems to validate that the control designed by student respects safety requirements. For that, they propose a validation filter

placed into the PLC.

In order to validate the approach, they applied it to a project with 10 year old children. The idea was to enable children to perform their first PLC program to control a large size packaging system called PRODUCTIS.

II. THE D.E.S TEACHING: FROM THEORY TO PRACTICE

Automatic control courses as all technical courses in the broad sense require the transfer of knowing and know-how to learners. In the case of the D.E.S teaching, the knowledge is characterized by the study, at different levels, of states automata, combinatory and sequential logics, Statecharts, Petri nets, Grafcet, SFC whose developments are still in progress [9], [10]. The level of knowledge is linked to the teaching level varying from discovery to specialization. Know-how concerns, for instance, the use and the programming of PLC by means of software respecting standard like IEC 1131.3 [11]. The acquisition of this technical know-how requires practical work in specialized and expensive rooms including PLC and simplified manufacturing systems which are a replica on a reduced scale of a real system found in the industry. These rooms moreover essential, are expensive, must be maintained by specialized personnel and are not generally in free access for security reasons. In this paper, they focus on the training use of operating industrial automation systems. They are interested in “large scale systems” with several inputs/outputs. That means that these systems can be decomposed in several sub-systems and have a high level of complexity. In addition, these systems are also able to perform several functions.

Hence, students use them as they would do it in their professional life. These systems are also subjected to hard tests by learners who can make errors of design in the control. In this paper, they focus on safety errors. This means that PLC outputs are not compatible with plant state. Safety errors can involve failures which make plant unavailable. Practical work with real plant requires, for the teachers, a lot of experiences, competences and time. It is important to note that teachers must generally manage about sixteen students simultaneously organized by pair. The development stations are not always located close to plant. While several students design or modify their control program, others are testing it on plant. Hence, at the time of the start-up, the teacher must supervise the plant; make sure that there are no errors in the controller and no failure of sensors and actuators, while managing the other students learning! It is more difficult in the case of a remote use. In this paper, they propose several solutions. When PLC and real plant are used, controller must be validated at least from the point of view of security before being implemented in the PLC. At Reims Champagne-Ardenne University, an automation system called PRODUCTIS is available. PRODUCTIS is an Integrated Manufacturing System which hinges around a pallet-based free transfer system as used in an industrial environment (Figure 1). It has been designed to bottle-pack medicine tablets. The system includes:

- two reference automatic subassemblies (small or large bottle) which distribute the tablets (white and green) through counting, close the bottle with a stopper and evacuate bottle,
- two equivalent removable automatic subassemblies (concept of subsystem) making it possible to carry out maintenance operations (disassembly/refitting, adjustment) in production conditions.

They have been designed to allow series changes (tools suited to two types of bottles and stoppers).

The process has been designed to carry out the following steps:

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

This system is composed of 2 PLC, 68 inputs and 33 outputs.

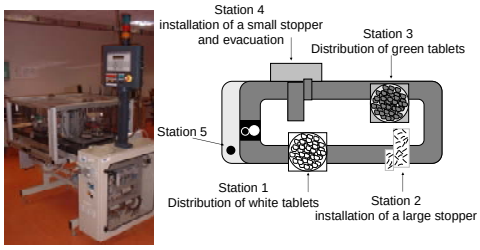

Figure 1. Productis

In this paper, they focus on the problem of logic controller design where students start from Running Specification Requirements (RSR) given by the teacher to propose a PLC implementation, whatever the programming tool, to control a real large scale system. The main problem for the teacher is to propose an exercise which is adapted to the student’s level. Next paragraph deals with the definition of “difficulty level of a logic controller design exercise” and how to modify and to adapt it to the student.

III. DIFFICULTY LEVEL OF A LOGIC CONTROLLER DESIGN PROBLEM

First, it is essential to define a logic controller design exercise. Usually, the control engineer divides the system into 2 parts: the Plant (P) and the Controller (C). The C observes the P state by means of sensors (E) and acts by means of actuators (S). A logic controller design thus consists to continuously determine the state of the output vector $S_i(t)$ according to the input vector $E_i(t)$ in order to match Running Specification Requirements. Given that the problems are seldom combinatory, a logic controller design can be formalized in the following way:

Find f respecting RSR such as $S_i(t) = f(E_i(t), S_i(t-1))$

Designing a logic controller necessarily requires a preliminary formalization stage of the Running Specification Requirements, also called specifications. The use of GRAFCET as a design methodology for logic controllers is increasing [12]. In this paper, they consider GRAFCET as the used specification tool [13], figure 2.

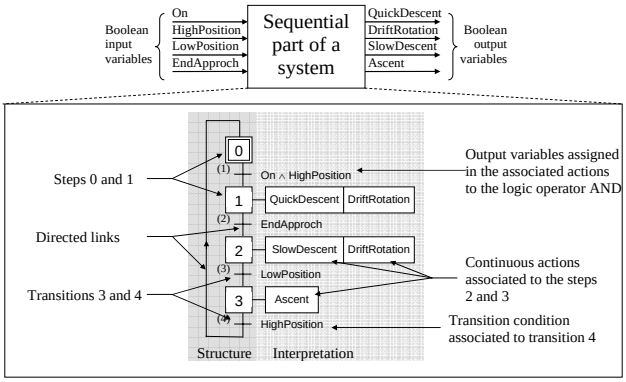

Figure 2. Structure and interpretation elements used in a GRAFCET chart to describe the behaviour of a sequential part of the system [13]

The stage of specification formalization requires an analysis of the Running Specification Requirements. Usually, the definition of the word "analysis" is the reduction of a complex element to several simple elements. The following stage is a synthesis stage, where specifications are transformed into logic program and placed into a PLC. For that, it is necessary to transform the GRAFCET into logical equations and to program them using IEC 1131-3. This international standard for programmable controller programming languages specifies the syntax, semantics and display for the following suite of PLC programming languages: Ladder diagram (LD), Sequential Function Charts (SFC), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL). GRAFCET can be easily converted into logic equations and then represented as ladder diagrams, for example. The method that describes more precisely the step behaviour is:

1) calculation of the Transition Functions (FTi)
2) calculation of the state variables (Xj)
3) calculation of the Outputs (Sk)

$$X_j(t) = FT_{PREDj}(t) \vee \left[X_j(t-1) \wedge FT_{SUCCj}(t) \right]$$

Where $FT_{PREDj}(t)$ are the preceding FT and $FT_{SUCCj}(t)$ are the following FT of the step *j* at time *t*.

It is obvious that the proposal and the definition for a control problem must be adapted to the learner's level. The analysis level, knowledge and competence required are not the same for a student who discovers the automatisms field and for a student who follows a specialization course. But whatever the level, to work on a real system is much more interesting and motivating for a learner. It is up to the teacher to define an exercise adapted to learner. However, a real model necessarily induces some constraints which considerably influence the difficulty level of a control problem. They try in the following paragraphs to clarify the parameters connected to the difficulty degree. Voluntarily, the "learner's point of view" and his/her perception of the difficulty level will not be considered.

A. Parameters linked to difficulty level

The concept of "difficulty" is quite close to the concept of "complexity". The characteristics of a "complex system" are: the high number and the large variety of variables, the big quantity of information, the significant number of subsystems, the interconnection between the subsystems… The perception of the system complexity, its analysis and its modelling are specific to the observer's objectives and his investigation and observation. For example, an atom is seen by everybody as an elementary particle and by the nuclear physicists as a complex system [14]. Morten Lind [15] considers that the systems can be broken up according to 2 axes called "Means-Ends" and "Whole-Part". By the distinction between means and ends, a system is, for Lind, described in terms of goals, functions and the physical components. At the same time, each of these descriptions can be given on different levels of whole-part decompositions.

They will show that this perception of a system can be used in their context. The level of difficulty (or the complexity) of the specification of a control problem, from their point of view, depends on 3 interdependent control parameters: the dimension, the hierarchization, and the synchronization.

1) Dimension parameter

The control dimension is directly related to the number of subsystems having to be controlled. It thus depends also on the number of sensors and actuators necessary to design the logic controller with regard to Running Specification Requirements. The larger the dimension is, the more important the effort is for learner, and the higher is level difficulty. That also means that the sequences of the Grafcet, will be necessarily longer.

2) Hierarchization parameter

Hierarchical control is directly linked to Running Specification Requirements. For example, the management of a "normal" cycle without taking into account the various operating modes does not require a hierarchized control. On the other hand, if the Running Specification Requirements are complete with respect to the operating modes, the specification will be more difficult and will require a hierarchical control structure. Once again, the analysis of the control problem will help to choose the right control structure (hierarchical structure, encapsulation, …).

3) Synchronization parameter

The solution of a logic controller design problem requires to synchronize some events and to coordinate several sequences. Simple synchronizations are the "selection of sequences" and the "simultaneous sequences". More complex coordination relate to the management of semaphores. Based on their experience in teaching, Figure 3 shows various synchronization types and their corresponding difficulty level, from "low" to "very high".

Cycle of a single sequence **(very low)**	Synchronization of sequences **(medium)**
Selection of sequences **(low)**	Synchronization and activation of parallel sequence **(medium)**
Step skip **(Low)**	IF event THEN… **(high)**
Backward sequence skip **(medium)**	common resources **(very high)**
Activation of parallel sequences **(low)**	Alternated sequences **(very high)**

Figure 3. Synchronization versus difficulty

B. Adaptation of difficulty level

The teacher can modulate the difficulty level of a logic control design by modifying either dimension, or synchronization, or structuration degrees inside Running Specification Requirements. The 3 parameters are not

independent of each other. To illustrate the subject, a pedagogical example is proposed (Figure 4).

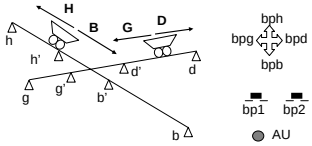

Figure 4 : Common zone

Let two carts, initially located, in *g* and *b* respectively share a common space. After pushing button *bp1*, cart 1 has to go to *h* and after goes back to *b*. After pushing button *bp2*, cart 2 has to go to *d* and then goes back to *g*. So, the control has to avoid collision. In addition, Running Specification Requirements includes an emergency mode. When the AU (emergency stop) button is pushed, the carts must go back to their initial positions by the means of a controlled manual mode (buttons *bpg*, *bpd*, *bph*, *bpb*). From the controller point of view, the inputs (E) are: AU, bp1, bp2, g, g', d', d, b, b', h', h, bpg, bpd, bph, bpb and the outputs (S) are G, D, H, B.

The difficulties of formalising Running Specification Requirements come from the common space management, that requires a specific synchronization, and the AU button management that imposes a hierarchical structure of control. To simplify logic controller design, the teacher can propose a Running Specification Requirements with only one cart or without the management of AU. The choice of the E/S makes it possible to decrease the degrees of synchronization and hierarchization. However, this approach of simplification acts only on the component level of the "Means-Ends" axis and only reduces the number of parts ("Whole-Part" axis). This decreases considerably the interest to use a real plant. They propose in the following paragraph another way to modify / adapt the difficulty level.

IV. Methodology to adapt difficulty level

The idea is to adapt the difficulty level by modifying Running Specification Requirements at the "functional" level of the "Means-Ends" axis. Hence by modifying the automation degree, it becomes possible to keep a global vision of the system. For that, they propose to adapt the difficulty level of Running Specification Requirements by using the functional dimension of the controller, and the autonomy given to the learner. These 2 aspects will make it possible to modify the automation degree.

To choose "the new" plant dimension the required teacher will be to define the inputs/outputs that the learner will be allowed to control. This work can be performed through a functional analysis of the plant. They propose the following representation of the functions. A function characterizes a sequence which can be more or less complex. A function thus integrates a degree of synchronization and structuration.

A function (Figure 5) is activated by the mean of a request for activation (RA) and is deactivated by the mean of a request for deactivation (RD). The effective engaging of the function can be made only if the activation conditions (Cai) are present. In the same way, the function deactivation is effective if the deactivation conditions (Cdi) are present. Fi1 characterizes the effective operation of the function. Fi2 represents the time between an activation request and a deactivation request. The function can be in autonomous mode or not. In the first case, the activation and the deactivation of the function will be done automatically when the activation and deactivation conditions are respectively true. In the contrary case, the learner has to activate or to deactivate the function at the right moment when the conditions are fulfilled. In this case, alarms (dsi, fsi) are set if the request does not coincide temporally with the conditions.

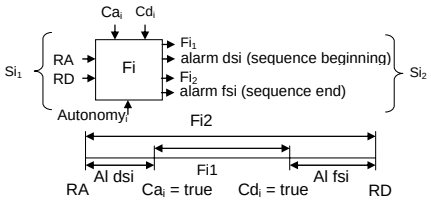

Figure 5. Function concept

The idea is to limit the perception of the plant and the possibilities of actions of the student. In other words, the student has to design a logic controller using advanced inputs/outputs called respectively AE1, AS1.

V. Controller validation

Work in the field of the automatic control validation aims to certify that mathematical properties are respected by the control model [16], [17], [18]. The work undertaken within the framework of tool UPPAAL [19] defines three types of properties: attainability, safety and liveness. In this work, they only consider "safety constraints": it is to say what the system should not do. The validation stage can be considered off line or on line. In the first case, the control is completely validated before being implemented into the controller [20]. 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 combinatory explosion, the difficulty to introduce the notion of "produced part" and to give a comprehensible explanation to the learner about his errors. In the second case, the validation is done in real time. This approach is complementary to those used in process supervision and fault diagnosis where the process state is compared to a dynamic model of the process [21]. They thus directed their work towards an on line approach of control validation, based on a validation filter established directly in the PLC. That makes it possible to be free from the asynchronous communication problems. By this approach of validation, the idea is to inhibit the evolutions which can lead the system to a situation of risk for operators and production resources. Cruette's work [22] for the monitoring of the automated systems proposes to intercalate a filter between the plant and the control. The filter ensures the coherence between the controller outputs and that which are expected, and the coherence between the evolution of the controller inputs and that which are expected with regard to outputs. This on line validation approach by filter

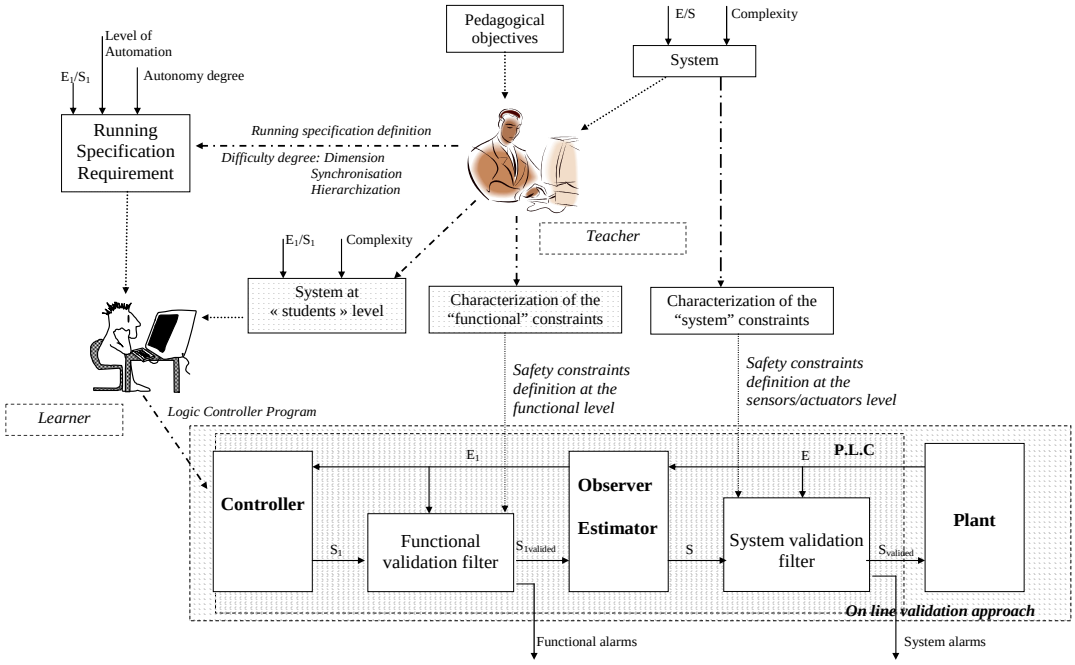

Figure 6. Validation Approach

is taken up partially and adapted to ensure the control validation (cf. Figure 6). The approach is based on 2 filters. A first sensor/actuators validation filter (also called "system validation filter") is at the plant level i.e. at new evolution of outputs S (actuators), the filter verifies that these ones are compatible with the plant state perceived by means of inputs E (sensors). However, the learner controls the plant with AE1 and AS1 placed at his disposal. A second functional validation filter makes it possible to valid coherence between the outputs AS1 and inputs AE1, and can generate alarms if the "autonomous" mode is selected. Only the "sensor-actuators" validation filter authorizes or not the sending of the S to the plant. If the order is validated by the filter, it is sent to the system, if not the system is stopped and the learner is informed. The functional validation filter reduces and defines the possible control errors coming from the student. It can also be useful to supply explanations concerning the error, but it is the sensors/actuators validation filter that guarantees the system safety. The 2 filters are placed in the PLC. It is necessary in addition to the 2 filters, to program the various functions in the PLC. This aspect is not detailed in this article. The following deals with the design of the 2 filters. Each filter contains constraints which have to be respected at each PLC cycle.

A. Functional validation filter

From the function model which has been proposed in paragraph IV, it is possible to write for each function the two following constraints: $RA \wedge Ca_i = 1 \qquad RD \wedge Cd_i = 1$

If the autonomous mode has been selected, that means that the learner has to design a control that respects the constraints. Alarms (dsi, fsi) are generated, if there is an error. If the autonomous mode has not been selected (by the teacher), the learner only controls the request to activate the function. In this case, functional constraints are not used. One can note that it is possible to define the possible accepted student's control by the mean of Activation and Deactivation conditions. Indeed, if for a function Fi, autonomous mode is selected and Cai is always true, it will be possible to detect that the function may be have not been activated at the right instant.

B. « sensors/actuators » validation filter

The definition of the safety constraints of the "sensor-actuators" validation filter is a difficult problem. To generate them automatically, behavioural plant models are necessary. Their approach is pragmatic and aims at proposing a classification of the various types of safety constraints. However, their definition must be made by the expert. It should be noted that this work is made only once because these constraints are valid for all the Running Specification Requirements relating to the plant. Methods like FMEA (Failure Modes and Effects Analysis) can be used to highlight the effects of control errors made by the student on the plant. They consider in this paper that the system states can be distinguished and modelled by the values of the Inputs (sensors) called uncontrollable states (*Xuc*) and Outputs (actuators) called controllable states (Xc) of the PLC. In other words, the system is supposed to be completely observable. The controller inputs (E) are called controllable events (Ec) for the sensors/actuators validation filter. In addition, the controller outputs (S) are named uncontrollable events (Euc). Two types of safety constraints are defined: the static safety constraints and the dynamic safety constraints.

1) Static safety constraints

The static safety constraints (SSC) express physical and technical impossibilities of the system elements. The static safety constraints depend only on controllable states. The Syntax is: $C = Xc_i \wedge Xc_j$. For example, if the command Xc_1 cannot be carried out at the same time as the command Xc_2, then: $Xc_1 \wedge Xc_2 = 0$.

2) Dynamic safety constraints

The dynamic safety constraints (DSC) relate to the

occurrence of an event which is not compatible with other event. 2 DSC are defined:

The combinatory DSC

The event corresponds either at the activation of a controllable event ($\uparrow Ec$) or an uncontrollable event ($\uparrow Euc$):

- In the first case, the constraint is written in the following way: $Xuc_j \wedge \uparrow Ec_j = 0$. Indeed, if the deactivation conditions are present, the sending of the associated controllable event is prohibited.
- In the second case, the constraint is written: $Xc_j \wedge \uparrow Euc_i = 0$. Indeed, as soon as the deactivation conditions are present, the actuator must be deactivated.

The sequential DSC

It is not always possible to express all the constraints as combinatory DSC because for that, it is necessary to have a sensor. If the sensor is not present, it is necessary to rebuild information. It is the case, for example, for the management of the common zone for the 2 carts example presented previously. The 2 carts are not allowed to be in the common zone at the same time. A possible solution, for the example, of the common resource between the 2 carts is proposed Figure 7. The 2 Grafcets respectively enable the horizontal cart position and the vertical cart position to be followed. In order to test the proposed approach, an original application with “novice control engineers” has been performed.

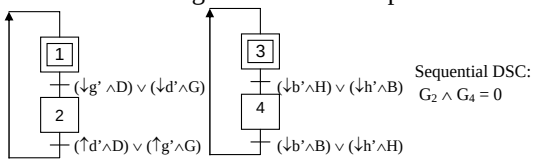

Figure 7 : DSC for the management of common zone

Safety constraints for the PRODUCTIS system have been completely designed and implemented in the PLC. Hence, students can program locally or remotely this system in a safety mode. The approach has been validated with students. The validation filter is implemented in the PLC and so used during practical courses. The validation filter corresponds to a specific module in the PLC containing all constraints and a test. At each cycle time, if one constraint is violated, the PLC output is not sent to the system and there is an alarm which is activated and displayed through SCADA software. The learners see the execution of the program running in real devices. If the program has logic/conceptual errors, the plant runs normally until a constraint is violated. After, the system is stopped.

VI. Original application with « novice control engineers »

The idea, in order to test the approach, was to propose to « novice control engineers », in their case 10-year-old children, to design their first logic control program to control the PRODUCTIS system. For that, they collaborated with a teacher of primary school. In the following paragraphs, choice of the level of difficulty and the control validation design stage are presented.

A. Definition of difficulty level

With regard to the age and level of the young control engineers, it was decided to decrease the level of difficulty at high rate. For that:

- Autonomous mode has not been selected,
- Component and functional dimensions have been reduced in order to decrease the numbers of inputs and to avoid control synchronization. In other words, the control program, for the children, is a cycle of a single sequence of functions.
- Only one function of the PRODUCTIS system can be active when a bottle is manufactured.

After functional identification of the system, they selected 20 functions (Table 1) that could be programmed by children. For that, they analysed the system by stations. The pallet is manually loaded (station 5). The child presses on a button to release the pallet. Each station is analysed here.

- Station 1: Distribution of green tablets and Station 3: Distribution of white tablets. Stations 1 and 3 performed two functions each other (F11, F31: distribute a tablet and F12, F32: release the pallet to go to the following station). The sequences generated by F11 and F31 are quite complex (backward sequence skip + selection of sequences). However, the modification of the functional dimension has completely withdrawn the complexity. Children control the distribution only by the mean of the output F11

Station 2: positioning of large stopper and Station 4: positioning of a small stopper and evacuation. This station is composed of a prehensor, i.e. two cylinders (one for the vertical movement and one for the horizontal movement), and a vacuum system. To install a stopper, it is necessary to place the cylinder to the top, go down, take the cap, go up, advance the cylinder, go down and release the aspiration. The functional identification is described at the lower level using the functions F21, F22, F23, F24, F41, F42, F43 and F44. In order to avoid synchronization in the control program designed by children, functions F25 and F45 (put the stopper) have been divided into two functions respectively: Take (F251 and F451) and Loosen (F252 and F452). With regard to the functional analysis, children also have to program the control of the ejection by means of the gripper (station 4) Through a FMEA, they decide that control errors would only be a bad activation of functions related to stations 2 and 4. For the 20 selected functions, activation (Ca) and deactivation (Cd) conditions can be found Table 1. One can note that a Ca can be equal to 1 in order to enable the system validation filter to detect several control errors (F_{21}: go out cylinder2, for instance).

The following paragraph deals with the design of the 2 validation filters.

1) Functional validation filter

When activation and deactivation requests are not synchronized with activation and deactivation conditions, this filter is able to generate *ds* and *fs* alarms. In this application, the non autonomous mode has been selected for all functions, so these constraints are useless.

2) System validation filter

In the proposed approach, security constraints are valuable whatever Running Specification Requirements. They have been obtained through an analysis (by the mean

Functional Identification				Ca	Cd	S1	S: PLC variables
Level 0	Level 1	Level 2	Level 3				
Packaging of tablets	P_1 : Distribute green tablets	F_{11} : Distribute a green tablet (1)		pallet in station1	tablet1	F11	Turn1+ : %Q2.18 Turn1- : %Q2.19
		F_{12} : Release the pallet to station1 (2)		pallet in station1	/pallet in station1	F12	Release1 : %Q2.16
	P_2 : Close a large bootle	F_{21} : Go out cylinder2 (3)		1	out2	F21	Go_out2 : %Q2.22
		F_{22} : Go in cylinder2 (4)		1	in2	F22	Go_in2 : %Q2.23
		F_{23} : Go up cylinder2 (5)		1	up2	F23	Go_up2 : %Q2.21
		F_{24} : Go down cylinder2 (6)		1	down2	F24	Go_down2 : %Q2.21
		F_{25} : Put the large stopper	F_{251} : Take2 (7)	1	$\uparrow F_{252-1}$	F251	Aspire2 : %Q2.48
			F_{252} : Loosen2 (8)	1	∅	F252	Aspire2 : %Q2.48 Eject2 : %Q2.49
		F_{26} : Release the pallet to station 2 (9)		pallet in station2	/pallet in station2	F26	Release2 : %Q2.17
	P_3 : Distribute white tablets	F_{31} : Distribute a white tablet (10)		pallet in station3	Tablet3	F31	Turn3+ : %Q2.34 Turn3- : %Q2.35
		F_{32} : Release the pallet to station 3 (11)		pallet in station3	/pallet in station3	F32	Release3 : %Q2.32
	P_4 : Close a small bottle or/and evacuate bottle	F_{40} : Close the small bottle	F_{41}: Go out cylinder 4 (12)	1	out4	F41	Go_out4 : %Q2.38
			F_{42} : Go in cylinder 4 (13)	1	in4	F42	Go_in4 : %Q2.39
			F_{43} : Go up cylinder 4 (14)	1	up4	F43	Go_up4 : %Q2.37
			F_{44}: Go down cylinder 4 (15)	1	down4	F44	Go_down4 : %Q2.36
			F_{45} : Put the small stopper F_{451} : Take4 (16)	1	$\uparrow F_{452-1}$	F451	Aspire4 : %Q2.50
			F_{452} : Loosen4 (17)	1	∅	F452	Aspire4 : %Q2.50 Eject4 : %Q2.51
		F_{46} : evacuate the bottle	F_{47} : Open the gripper (18)	1	∅	F47	Open : %Q2.25
			F_{48} : Close the gripper (19)	1	∅	F48	Close: %Q2,24
		F_{49} : Release the pallet to station 4 (20)		pallet in station4	/pallet in station4	F49	Release4 : %Q2,33

Table 1: functional identification of Productis machine

of a specific FMEA) of the consequences on the PRODUCTIS of control errors.

In this paper, they only indicate security constraints which can be not respected because of a bad control program.

Static safety constraints

There are not SSCs because only one function of the PRODUCTIS system can be active when a bottle is manufactured.

Combinatory dynamic safety constraints

$\uparrow$GO_OUT2$\wedge$out2=0 (1) $\uparrow$RELEASE1$\wedge$up2=0 (9) $\downarrow$ASPIRE4$\curlywedge$(down4$\wedge$out4)=0 (17)
$\uparrow$GO_IN2$\wedge$in2=0 (2) $\uparrow$GO_OUT2$\wedge$up2=0 (10) $\uparrow$CLOSE4$\curlywedge$(down4$\wedge$in4)=0 (18)
$\uparrow$GO_UP2$\wedge$up2=0 (3) $\uparrow$GO_OUT4$\wedge$up4=0 (11) $\uparrow$ASPIRE2$\curlywedge$(down2$\wedge$in2)=0 (19)
$\uparrow$GO_DOWN2$\wedge$down2=0(4) $\uparrow$RELEASE2$\wedge$up2=0(12) $\downarrow$ASPIRE2$\curlywedge$(down2$\wedge$out2)=0(20)
$\uparrow$GO_OUT4$\wedge$out4=0 (5) $\uparrow$RELEASE3$\wedge$up4=0 (13) $\uparrow$GO_IN2$\wedge$ up2= 0 (21)
$\uparrow$GO_IN4$\wedge$in4=0 (6) $\uparrow$RELEASE4$\wedge$up4=0 (14) $\uparrow$GO_IN4$\wedge$ up4 = 0 (22)
$\uparrow$GO_UP4$\wedge$up4=0 (7) $\uparrow$OPEN4$\curlywedge$(down4$\wedge$out4)=0 (15)
$\uparrow$GO_DOWN4$\wedge$down4=0 (8) $\uparrow$ASPIRE4 $\wedge$ /(down4 $\wedge$ in4) = 0 (16)

Sequential dynamic safety constraints

There is not a sensor enabling to know if the grip is open or closed. So, it is necessary to construct a state estimator (Figure 8).

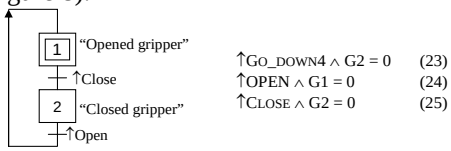

Figure 8 : State estimator of gripper position

B. *Activity with children*

The activity with the children proceeds in two steps. In the first, the child has at his/her disposal an HMI (Human-Machine-Interface) 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. For that, the child clicks a button and the associated function starts. According to the state of the system, all the buttons are not activated. For example, if the cylinder of station 2 is in position "in2", the button "To Go_in the cylinder" of station 2 can not be clicked (no entry sign on the button). This button is inactive until the cylinder is in the position "out". After having understood the function behind each button, the child can perform the second part of work (second HMI).

During the second activity the child programs his own sequence of functions to bottle medicine tablets through a specific Human-Machine Interface. When the sequence is considered as correct by the child, it is sent to the PLC and the sequence execution is validated on line. The running of the PRODUCTIS system is displayed on a multimodal interface [23]. 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.

Let them suppose that the child proposes the following sequence ($F_{12} \rightarrow F_{24} \rightarrow F_{251} \rightarrow F_{21} \rightarrow F_{252}$) knowing that there is a pallet at station 1, the cylinder of station 2 is in "up and in" position (*up2*=1, *in2*=1) and the vacuum is not active (*Aspire2*=0). During the sequence execution, the function F12 generates the output *Release1* (%Q2.16) and the constraints set is respected. The output is then sent to the system which effectively releases the pallet at station 1. It is the same for the functions F24 and F251 which generates outputs: Go_down_2 and $Aspire_2$. The control error comes when the function F21 is activated. In this case, the output Go_out_2 becomes equal to 1 and the constraint (10) is not validated. So, the PRODUCTIS is stopped and the validation system informs the child of his error. Afterwards, he/she must modify his/her control sequence and he/she starts the system to validate it again. This activity has had a great success and most of the children performed the control programming task.

VII. CONCLUSION

This paper dealt with remote use of operating industrial automation system for training in D.E.S field. The 2 main ideas are:

- To adapt the difficulty level of logic controller design. For that, they propose to modify the level of automation without changing the size of the manufacturing system. The principle consists of proposing to the student "Running Specification Requirements" at a "functional" level. Hence, it becomes possible to keep a global vision of the system. A "function" model adapted for that has been proposed.

- The design of 2 validation filters in order to guarantee the safety. One filter called "system validation filter" validates outputs before sending them to the plant. This filter is based on logical constraints which are classified in SSC, Combinatory and sequential DSC. The second filter called "functional validation filter" validates the use of the functions with regard to the autonomy mode selected. In fact, this filter reduces the use of safety constrains which could be violated in the system validation filter. This approach has been validated with "young novice control engineers" who designed their first control program on a real operating industrial automation system called PRODUCTIS which bottle-packs medicine tablets. This work can have several interesting perspectives. First of all, in the field of remote or e-maintenance, the validation filters can be used in order to guarantee the safety of operators and materials [24]. Secondly, they intend to propose a remote use (through Internet) of their automation systems to schools in order to enable young people to discover the automation field. At least, they are now working on the validation of liveness specification in order to be able to check the full logic controller designed by a student for specific Running Specification Requirements.

REFERENCES

[1] Sim K.B., Byun K.S., Harashima F., Internet based tele-operation of intelligent robot with optimal 2 layer fuzzy controller, *IEEE Trans. on Indus. Elec*.,vol. 53, Issue 4, Page(s):1362 - 1372 June 2006

[2] Marangé P., Gellot F., Chemla J.P., Riera B., "Requirement and Use for remote teaching of Discrete Events Systems", *Proceeding of 7th IFAC symposium on Advances in control Education, ACE'06*, Paper WeP02.1sur le CD-ROM, Madrid, Spain, June 21-23, 2006

[3] Metzger M., "Agent-based virtual control systems for DCS education via Internet", S3b_4, *Second IFAC Workshop on Internet-based control education*, Grenoble, Sept. 2004, CD-edition.

[4] Lunt B.M., Helps H.G., Carter P., Red E., "Systems and automation education through Web-based labs", *ICEE'00*, Taiwan, august 2000

[5] Colace F., De Santo M., Pietrosanto A., "Work in Progress - Virtual Lab for Electronic Engineering Curricula", *34th ASEE/IEEE Frontiers in Education Conference*, October 22 – 24, 2004, Savannah,

[6] Muškinja N., Tovornik B., "Swinging up and Stabilization of a real interved pendulum", *IEEE Trans. on Indus. Elec*.,vol. 53, n°2, Page(s):631 - 639 April 2006

[7] Haddapis G., "An interactive electronic book approach for teaching computer implementation of industrial control systems", *IEEE Trans. on Educ.*, vol.46, n°1, February 2003

[8] Gomis Bellmunt O., Montesinos Miracle Daniel, Galcern Arellano S., Sudria Andreu A., "A distance PLC programming Course employing a remote laboratory based on a flexible manufacturing cell", *IEEE Trans. on Educ.*, vol.49, n°2, Page(s):278 – 284,may 2006

[9] Golmakani H., Mills J., Benhabib B., "Deadlock-free scheduling of flexible manufacturing workcells using automata theory", *IEEE trans. on systems man and cybernetics*, vol. 36, n° 2, Page(s):327 – 337, march 2006

[10] Polic A., Jezernik K., "Closed-loop Matrix based of discrete event system for machine logic control design", *IEEE Trans. on Indus. Infor.*, vol.1, n°1, p39-46, February 2005

[11] International Electrotechnical Commission, "Preparation of function charts for control systems, International Standard", *CEI/IEC 848*, 1991 (revised version).

[12] 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, Spain, June 21-23, 2006

[13] International Electrotechnical Commission, "*Preparation of function charts for control systems*". Publication 848, 2002

[14] Modarres, M., Cheon, S-W., "Function-centered modeling of engineering systems using the goal tree-success tree technique and functional primitives", *Reliability Engineering & System Safety*, Volume 64, Issue 2, May 1999, Pages 181-200.

[15] Lind, M., "Modeling Goals and Functions of Complex Industrial Plant. Applied Artificial Intelligence", Vol8 No.2,April-June 1994

[16] Emerson E.A., "Temporal and modal logic", *in Van Leeuwen D.J, editor. Handbook of the theoretical Computer Sciences*, vol. 9, chapter 16, pages 995-1072, Elsevier Science Publishers and the MIT Press1990

[17] Pollmacher D.; Zimmermann W.; Hanisch H.-M, "Translation validation for model-based code-generators for PLCs. "; *Emerging Technologies and Factory Automation, 10th IEEE Conference* on Volume 1, Page(s):8 pp. 19-22 Sept. 2005

[18] 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

[19] Behramm G., David A., Larsen K.G., "A tutorial on UPPAAL", novembre 2004

[20] Marangé P., Tajer A., Gellot F., Carré-Ménétrier V., "Synthesis of supervised controller based on Boolean constraints and Boolean automata", *INCOM'2006: 12th IFAC/IFIP/IFORS/IEEE/IMS Symposium Information Control Problems in Manufacturing*, may 17-19 2006, Vol. 1, p299-304, St Etienne, France

[21] Lo, C. H., Wong, Y.K. and Rad, A.B, "An Intelligent System for Process Supervision and Fault Diagnosis in Dynamic Physical Systems", *IEEE Trans. on Indus. Elect.*, Vol. 53 (2), April 2006, pp. 581-592, 2006.

[22] 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.

[23] Marin, R.; Sanz, P.J.; Nebot, P.; Wirz, R., "A multimodal interface to control a robot arm via the web: a case study on remote programming", *IEEE Trans. on Indus. Elect.*, Vol.52, Iss.6, Pages: 1506- 1520, Dec. 2005.

[24] Matsumoto, Y.; Katsura, S.; Ohnishi, K., "Dexterous Manipulation in Constrained Bilateral Teleoperation Using Controlled Supporting Point", *IEEE Trans. on Indus. Elect.*, Vol.54, Iss.2, Pages:1113-1121, April 2007.

Marangé Pascale is currently working toward the Ph.D. degree in Automatic Control at the University of Reims Champagne Ardenne (URCA), France. Her research interests include the verification and validation of PLC program, for the remote use in the case of teaching or remote maintenance.

Gellot François received the Ph.D. degree in Automatic Control from the University of Reims Champagne Ardenne (URCA), France. He is an Associate Professor of Control Engineering at the University of Reims Champagne-Ardenne (URCA) France, and a Researcher at the CReSTIC (research center in Sciences and Technologies on Information and Communication). His research interests include the modelling, analysis, synthesis and validation of Discrete Event Systems.

Riera Bernard received the Ph.D. degree in Automatic Control from the University of Valenciennes (UVHC), France, in 1993. He is a Professor of Control Engineering at the University of Reims Champagne-Ardenne (URCA) France, a Researcher at the CReSTIC (research center in Sciences and Technologies on Information and

Communication) and head of the “automation and hybrid systems” team. His research interests include supervisory control of hybrid systems, supervisory support systems, discrete events and hybrid systems modelling.