Application of the Control Validation to the D.E.S. Teaching

Pascale Marangé, François Gellot, Bernard Riera
CReSTIC
Moulin de la housse / BP1039 / 51687 REIMS cedex2 / FRANCE

***Abstract*—Objectives of Automation courses are knowledge and know-how transfer to students. It is important for learner to control locally or remotely real systems 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. Technologies today allow 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, we 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 logical constraints, which should in no case to 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. In order to validate the approach, we applied it to an original project with 10-years-old children on a packaging system.**

***Index Terms*—Discrete events system, validation, control, functional identification, remote laboratories, DES teaching.**

I. INTRODUCTION

Nowadays, the use of Communication and Information Technologies is a reality in the automation areas. Indeed, one can find a massive use of the Ethernet network, as well on 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 able 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 automation, automatic control and Discrete Events System (D.E.S) theory. The idea that we 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 areas of the automatic control of continuous processes, use of virtual and/or remote laboratories for teaching is well known. We 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 feedback control can be found in relevant literature [4] [5] [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, we propose an original solution to solve these two problems. For that, on the one hand, 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 to be violated. On the other hand, one of main difficulty is to adapt the plant system to the different users, keeping the device as a whole. We define in the paper the difficulty level of a Logic Controller Design (LCD) by means of three 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, we propose to modify the functional dimension of the plant and the student's level of autonomy.

In order to validate the approach, we applied it to a project with 10-years-old children. The idea was to enable children to perform their first PLC program to control a large size packaging system called Productis. 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 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 and station 3), bottle closing (station 2 and station 4), bottle evacuation (station 4).

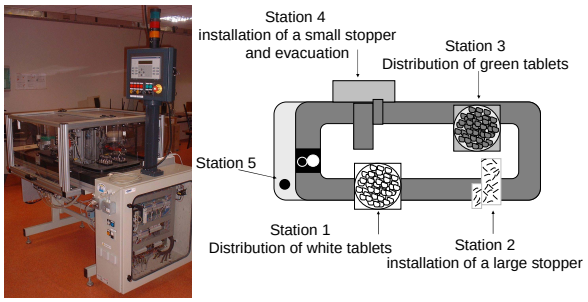

Figure 1. Productis

II. THE DES TEACHING: FROM THEORY TO PRACTICE

Automatic control 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 61131.3 [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, we focus on the training use of operating industrial automation systems. 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. Practical work with real plant requires for the teachers a lot of experiences, competences and time. During the controller test, the teacher must supervise the plant; make sure that there are no error in the controller and no failure of sensors and actuators. It is more difficult in the case of a remote use.

In this paper, we 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 (i.e. several inputs/outputs). 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 student.

A. *Difficulty level of a LCD problem*

First, it is essential to define a logic controller design problem. From a practical point of view, 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 RSR. Being 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 RSR, also called specifications. The use of Grafcet as a design methodology for logic controllers is increasing [12]. In this paper, we consider Grafcet as the used specification tool [13].

The stage of specification formalization requires an analysis of the RSR. Usually, the definition of the word "analysis" is the reduction of a complex element to a 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 international standard for programmable controller programming languages using IEC 61131-3 [11].

It is obvious that 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 areas 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 to the teacher to define an exercise adapted to learner. We try in the following paragraphs to clarify the parameters connected to the difficulty degree.

1) *Student control errors*

Student can make mistakes during the control design stage. These errors can be classified as following: syntaxical errors and specification errors. We are not interested in syntaxical errors because they will be detected during the programming stage by the PLC software. Specification errors can have different consequences on the plant. For instance, an error can involve the plant either to a state which does not correspond to the specification or to a forbidden state which is very dangerous. In this paper, we only want to avoid safety consequences of specification errors. Necessary, errors come from a "Bad" command sent by the PLC. "Bad" means in this case, not adapted to the context of the production system. In our approach, we model the context through the system state.

2) *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. Morten Lind [14] 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. We use this perception of a system in our context. The level of difficulty of the specification of a control problem, from our point of view, depends on 3 interdependent control parameters: the dimension, the hierarchization, and the synchronization. The teacher can modulate the difficulty level of a logic

control design by modifying either dimension, or synchronization, or structuration degrees inside RSR. The 3 parameters are not independent to each other. The choice of the E/S makes possible to decrease the degrees of synchronization and hierarchization. We propose in the following paragraph another way to adapt the difficulty level.

B. Methodology to adapt difficulty level

The idea is to adapt the difficulty level by modifying RSR 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, we propose to adapt the difficulty level of RSR 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. 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. To choose "the new" plant dimension requires for the teacher to define the inputs/outputs. This work can be performed through a functional analysis of the plant. We 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 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. Fi_1 characterizes the effective operation of the function. Fi_2 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.

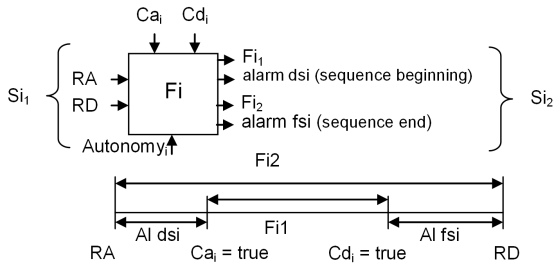

Figure 2. Function concept

III. CONTROLLER VALIDATION

Work in the areas of the automatic control validation aims to certify that mathematical properties are respected by the control model [15], [16], [17]. The work undertaken within the framework of tool UPPAAL [18] defines three types of properties: attainability, safety and liveness. In this work, we only consider "safety constraints": it is to say what the system should not do. 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 [19]. Our work towards an on-line approach of control validation, based on a validation filter established directly in the PLC.

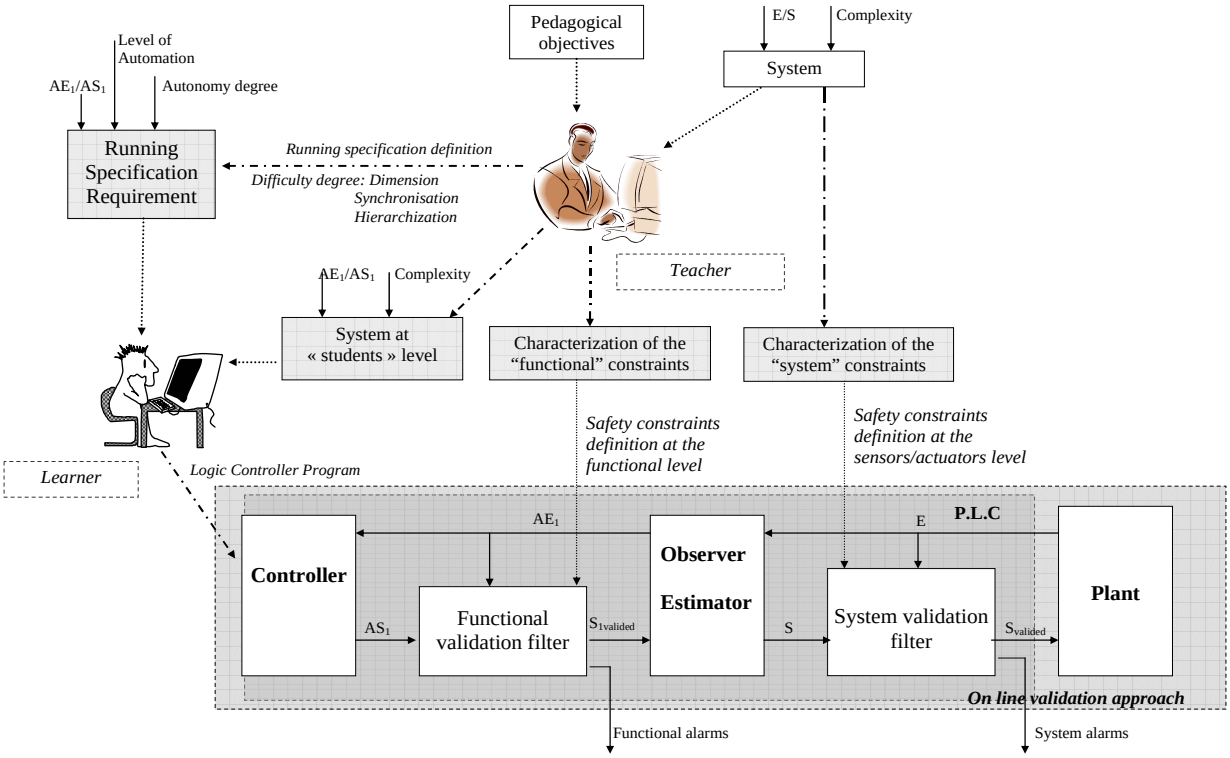

Figure 3. Validation Approach

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 [20] for the monitoring of the automated systems proposes to intercalate a filter between the plant and the control. This on-line validation approach by filter is taken up partially and adapted to ensure the control validation (figure 3). The approach is based on 2 filters. A first "system validation filter" is at the plant level i.e. at new evolution of outputs S (actuators), the filter verifies that these one 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 "system 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 (Observer/Estimator). This aspect is not detailed in this article. The following of the paper deals with the design of the 2 filters.

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 (Ca_i and Cd_i). Indeed, if for a function Fi, autonomous mode is selected and Ca_i is always true, it will be possible to detect that the function may be has 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, the expert must make their definition. 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 events. Two 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. 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 4. 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.

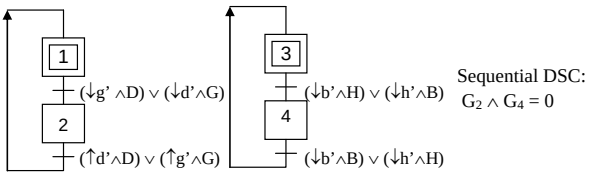

Figure 4: DSC for the management of common zone

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.

3) Approach to obtain the constraints

We consider in this paper that the system states can be distinguished and modeled by the values of the Inputs (sensors) and Outputs (actuators) of the PLC. In other words, the system is supposed to be completely observable. We distinguish the controllable state (outputs, called *Xc*) from the uncontrollable state (inputs called *Xuc*). The inputs enable to know either the actuators positions or the products positions. In this paper, we only consider actuators locations. In order to avoid combinatory explosion, a modular approach is necessary. So, firstly the full system has to be independently modelled through the different actuators. Secondly, interactions between actuators have to be studied. Hence, the system is modelled by the mean of two models by actuator, one representing the different position of the actuator and one for the different possible outputs. In addition, one model is built by interaction between plant actuators. In this approach, finite states automata are used as modelling tool. It exists 3 different system states: (i) an authorized state is a state which is always accepted. (ii) a forbidden state has always to be avoided. That means that a forbidden state is reached by a change of an output (called controllable event: *Ec*). (iii) a fugitive state is a state which can not be avoided but which has to be left as soon as possible. That means that a fugitive state is reached by a change of an input (called uncontrollable event: *Euc*).

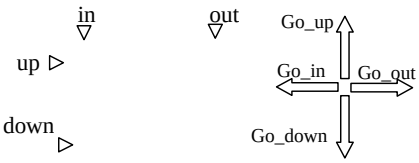

Figure 5. Prehensor

The approach is illustrated by the mean of the station 2 extracted from the Productis system. It is the prehensor (figure 5) which is composed of two cylinders, a horizontal cylinder {Xuc: *in*, *out*; Xc: *Go_in Go_out*} and a vertical cylinder {Xuc: *up*, *down*; Xc: *Go_up*, *Go_down*}.

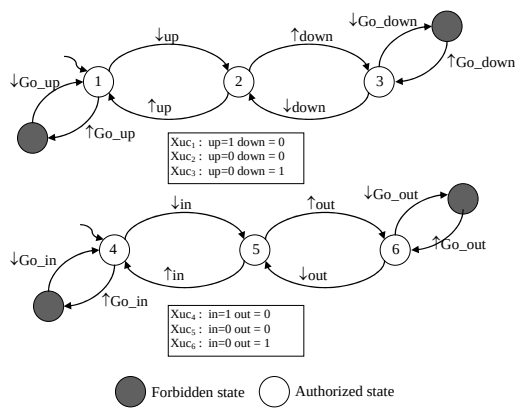

Figure 6. Cylinders position automata

Firstly, we consider the plant element independently. Each prehensor can have 3 positions which correspond to 3 uncontrollable states and it exists for each 2 forbidden states which can be reached by a controllable event. For instance, when the vertical cylinder is up; it is not allowed to "Go up" it. Figure 6 shows the corresponding automata model. From the model, one can define logical equations to implement in the PLC in order to detect if a forbidden state has been reached: $Xuc_i \wedge Ec_i = 0$ (1)

The models representing the different outputs of the 2 cylinders are proposed figure 7. In this case, for each actuator, they are one forbidden state (2 outputs activated) and two fugitive states. For instance, the cylinder has to be stopped when it is out. As previously, it is also possible to define logical equations to implement in the PLC, which enable to detect if a forbidden state has been reached or if a fugitive state has not been left:

$Xc_i \wedge Ec_i = 0$ (2) $Xc_i \wedge Euc_i = 0$ (3)

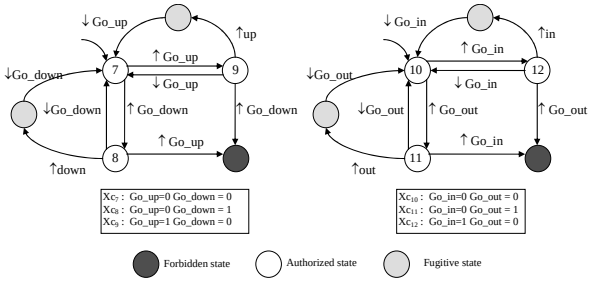

Figure 7. Cylinders output automata

Secondly, the interaction between the 2 cylinders is modelled. The model of the different positions is built by asynchronous product between the position automatons. In this case, there are 9 positions. For security reasons, the positions 4 and 7 are forbidden (figure 8).

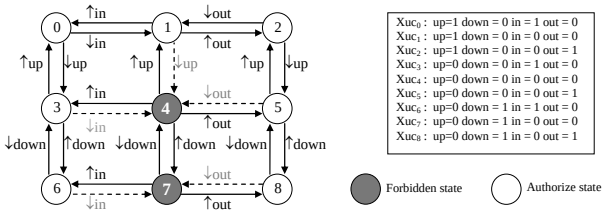

Figure 8. Prehensor Position automaton

It is possible from these models and some precedence and occurrence relations to generate constraints, which can be implemented in the PLC. This part is not described here.

IV. ORIGINAL APPLICATION

Our idea in order to test the approach was to propose to « novice control engineers », in our case 10-years-old children, to design their first logic control program to control the Productis system. For that, we collaborate 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 has been decided to decrease a lot the level of

difficulty. 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 is a cycle of a single sequence of functions,
- Only one function can be active.

After functional identification of the system, we selected 20 functions (extract in table 1) that could be programmed by children. For that, we 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 after.

TABLE I.
FUNCTIONAL IDENTIFICATION OF PRODUCTIS MACHINE

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

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; 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. These stations are composed of a prehensor, i.e. two cylinders, 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 vacuum. 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 respectively two functions: Take (F251 and F451) and Loosen (F252 and F452). Through a FMEA, we 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.

B. *Activity with children*

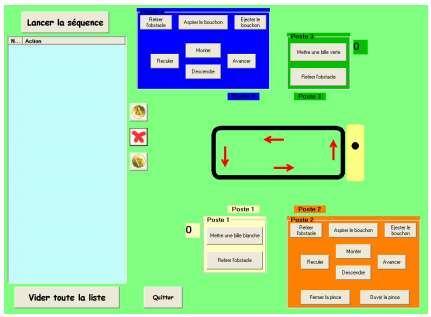

Figure 9. Human-Machine Interfaces : "Step by step" mode

The activity with the children proceeds in two steps. In the first, the child has at his/her disposal an HMI (Human-Machine-Interface) (figure 9) 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, not all the buttons are 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 (figure 10), 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.

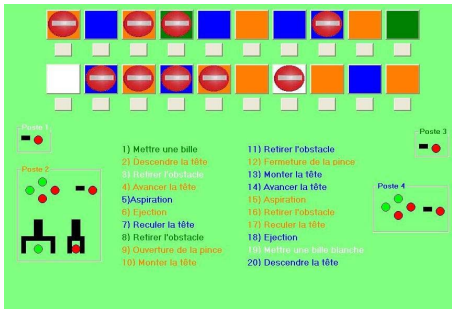

Figure 10. Human-Machine Interfaces : Sequence mode

V. CONCLUSION

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

To adapt the difficulty level of logic controller design. For that, we 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 areas of remote or e-maintenance, the validation filters can be used in order to guarantee the safety of operators and materials. Secondly, we intend to propose a remote use (through Internet) of our automation systems to schools in order to enable young people to discover the automation areas. The approach is going to be extended to partially observable system. At least, we 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 RSR.

ACKNOWLEDGMENT

The research reported here has been supported by the Champagne-Ardenne countryside.

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, *IFAC Workshop IBCE'04*, Grenoble, France, September 5-7 2004

[4] Lunt B.M., Helps H.G., Carter P., Red E., "Systems and automation education through Web-based labs", *ICEE 2000*, Taiwan, august 14-16, 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 20–23, 2004, Savannah, GA

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

[15] Emerson E.A., Van Leeuwen D.J., "Temporal and modal logic", *Editor Handbook of the theoretical Computer Sciences*, vol. 9, chapter 16, pages 995-1072, 1990

[16] Pollmacher D.; Zimmermann W.; Hanisch H.-M, "Translation validation for model-based code-generators for PLCs", *ETFA'05 10th IEEE Conference*, on Volume 1, pp. 19-27 September 2005

[17] 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

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

[19] Lo. C.H., Wong Y.K., Rad A.B., "Intelligent System for Process supervision and fault diagnosis in dynamic physical systems", *IEEE Trans. on Indus. Elec.*, vol.53, n°2, p581-589, April 2006

[20] 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* 1998

AUTHORS

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. (pascale.marange@univ-reims.fr)

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. (francois.gellot@univ-reims.fr)

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. (Bernard.riera@univ-reims.fr)

Manuscript received 10 January 2008. Published as submitted by the authors.