MODELLING AND IMPLEMENTING THE CONTROL OF AUTOMATED PRODUCTION SYSTEMS USING STATECHARTS AND PLC PROGRAMMING LANGUAGES

José J.B. Machado, Farid Louni, Jean-Marc Faure, Jean-Jacques Lesage, Jaime Fereira, Jean-Marc Roussel

MODELLING AND IMPLEMENTING THE CONTROL OF AUTOMATED PRODUCTION SYSTEMS USING STATECHARTS AND PLC PROGRAMMING LANGUAGES

José M. MACHADO [(1)] / Farid LOUNI [(2)] / Jean-Marc FAURE [(2)] / Jean-Jacques LESAGE [(3)] / Jaime C. L. FERREIRA DA SILVA [(1)] / Jean-Marc ROUSSEL [(3)]

(1) *Mechanical Engineering Department*
School of Engineering, University of Minho
Campus de Azurém, 4800-058 Guimarães, PORTUGAL
Tel: +351 253 510220 – Fax: +351 253 516007
Email: jmachado@eng.uminho.pt; jfs@eng.uminho.pt

(2) *ISMCM - CESTI*
Département Production-Automatique
3 rue Fernand Hainaut
93407 Saint-Ouen Cedex – FRANCE
Tel: 33 (0)1 49452900 – Fax: 33 (0)1 49452991
Email: flouni@ismcm-cesti.fr; jean-marc.faure@ismcm-cesti.fr

(3) *Ecole Normale Supérieure de Cachan*
Laboratoire Universitaire de Recherche en Production Automatisée (LURPA)
61, Avenue du Président Wilson - 94235 Cachan Cedex - FRANCE
Tel: 33 (0)147402218 - Sec. 33 (0)147402215 - Fax. 33 (0)147402220
Email: lesage@lurpa.ens-cachan.fr; jean-marc.roussel@lurpa.ens-cachan.fr

Keywords: Statecharts, IEC 61131-3 languages, Discrete Events Systems, Automated Production Systems.

Abstract

This paper is concerned with the applicability of statecharts for the discrete control of production systems.
We show first how it is possible to build a control model described with the statecharts formalism and taking into account :

- The operating modes and the physical structure of the system,
- The behaviour of different kinds of actuators.

Then we present a method enabling one to obtain a PLC program in IEC 61131-3 languages from the statecharts model previously elaborated.
An application example illustrates these two points.

1 - Introduction

The classical V-shaped model of the life-cycle of an Automated Production System (APS) (Figure 1) is composed of several phases. The work presented in this paper focuses on the points 4 (conception of the control part) and 5 (implementation of the control part). During each of these two phases, a model is to be built. The conception phase aims at elaborating an accurate control model to describe all the possible evolutions of the system. The objective of the implementation phase is to produce a controller program compliant with the previously elaborated control model.

With concern for standardisation, it is obvious that IEC 61131-3 languages must be used for the controller program. On the other hand it is not so easy to choose the most suitable language for control model elaboration. Several languages like Grafcet, control Petri nets, statecharts, are candidates. All these languages are based on state machines (Moore machine, Mealy machine). For sequential systems with parallelism, selection, rendez-vous mechanisms, Grafcet [IEC 60848] and control Petri nets are very powerful languages. This explains their popularity with control engineers.

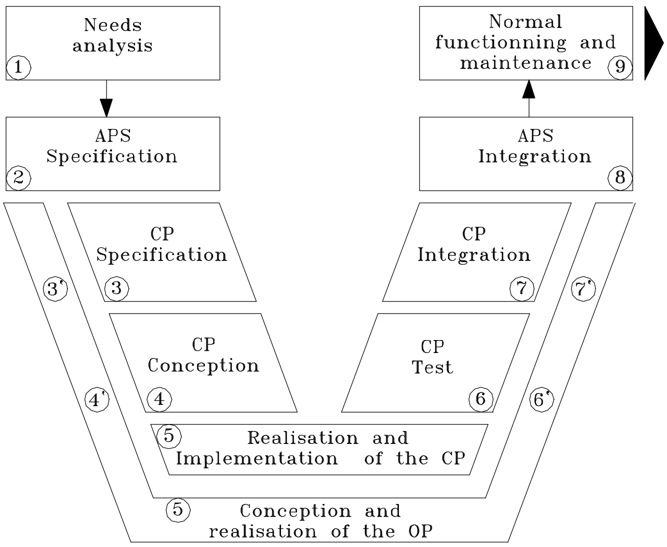


Figure 1) Automated Production System life cycle.
APS – Automated Production System
CP – Control Part
OP – Operative Part

The diagrams proposed by David Harel [Harel.87], called statecharts, are based on an extended state-transition formalism which also allows one to express parallelism, concurrency and communication between states. Its most interesting characteristic from our viewpoint is the concept of state hierarchy (some states can be refined and include sub-states). Software environments, like STATEMATE, STATEFLOW, enabling one to design statecharts models are to-day available and several papers dealing with statecharts semantics [Harel.96] or presenting application cases have been published. Nevertheless, and as far as we know, no application to a real automated production system with several tens input/output control variables and real actuators control has been presented. Moreover no method enabling one to structure a statecharts model has been given.

When building the control model of a real production system, it is however necessary:

- To have at one's disposal some structure criteria
- To be able to express the control of real actuators, in order to generate physical control outputs.

So the aim of our study is to evaluate the possibility to use statecharts when modelling the control of a real production system. More precisely we intend :

- To develop a method enabling us to structure a statecharts model according to relevant criteria
- To test the potentiality of statecharts for actuators control description
- To develop a method of translation of a statecharts model into a IEC 61131-3 program.

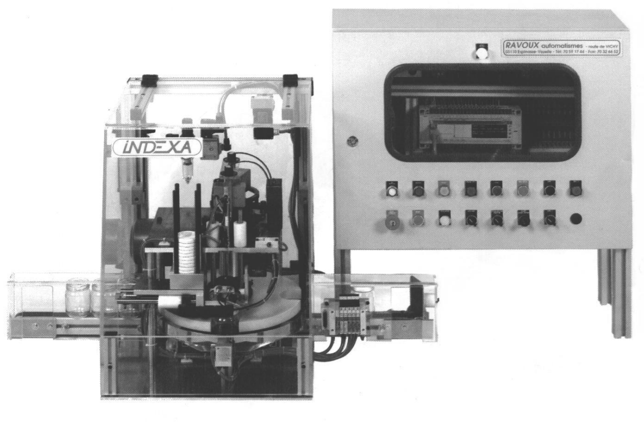
The paper is organised as follows: We present in chapter 2 the method developed in order to elaborate the control model. This method comprises two phases. First, a state hierarchy is defined based on the operation modes of the production system and on its physical structure. Then, all the states of this hierarchy are refined; their refinement depends on the actuators` technology. We obtain at last a statecharts model able to control all the outputs. In chapter 3 is presented the translation method from a statecharts model into a PLC program. The conclusions and the perspectives of this study are presented in chapter 4.

This research project is the result of the co-operation between three institutions: Mechanical Engineering Department of University of Minho in Portugal (DEM), ISMCM – CESTI Paris (Institut Supérieur des Matériaux et de la Construction Mécanique – Centre d'Etudes Supérieures des Techniques Industrielles) in France and LURPA (Laboratoire Universitaire de Production Automatisée) from ENS (Ecole Normale Supérieure de Cachan) also in France.

2 – Model Elaboration

The equipment used for the present work is an automated production machine belonging to the LURPA and named INDEXA. This equipment is a typical automated production system. So the obtained conclusions, either for the statecharts model structure or for the PLC program building, can be adopted for any other automated production system control.

The purpose of this machine (see figures 2 and 3) is to place



covers on jam jars.
Figure 2) INDEXA machine

The normal operation mode of this machine is the following:

- The jam jars are carried to the initial position by a conveyor actuated by an electric motor. This position is one of the four stable positions of a table that rotates with a "croix de malte" mechanism.
- The second position is the work station, where the covers are conveyed and screwed upon the jars, by combined movements of an horizontal cylinder, a vertical cylinder and a pneumatic rotation head (rotation movement). A cover warehouse is also available at this table position.

- The third position is the output of the jars with covers, using the same conveyor as the first position.
- The fourth position is empty.
- The rotation of the table is performed by a “croix de malte” mechanism driven by an electric motor. The table’s rotation carried out with a “croix de malte” mechanism enables a high precision positioning and the use of a single conveyor. (Positions 1 and 3 of the table).

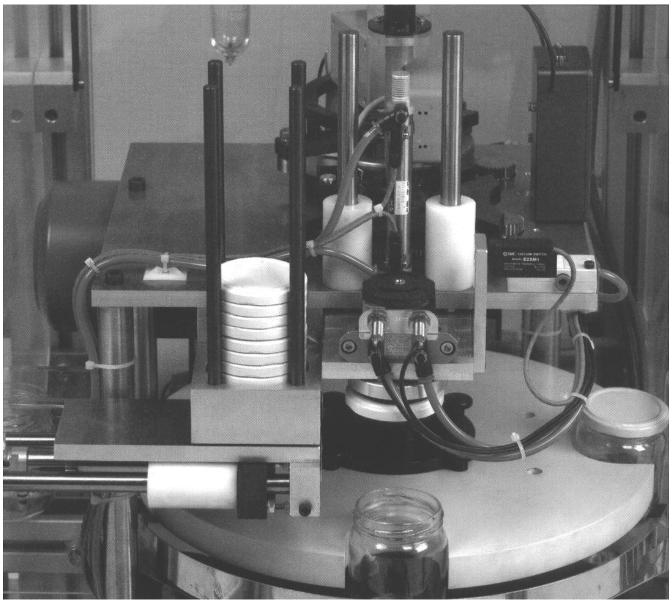


Figure 3) INDEXA`s workstation.

This normal operation mode can be executed:

- Manually. The movement from a position to the following one requires an action of the operator.
- Automatically.

Moreover, three others states, corresponding to operation modes without added value, but mandatory for safety reasons, have been defined.

- Emergency Stop: machine without electric and pneumatic energy. This state is activated by pushing on the emergency bottom or by opening the physical protections of the machine.
- Stand by: This state is an intermediate state between the “Without Energy” state and the “Initial” state. To reach this state from the “Without energy” state the operator has to push the “Energy” button; to leave this state he has to push the “Initialisation” button. All the actuators go then automatically to their initial position.
- Initial state: obtained after initialisation of the machine. All the actuators are ready to operate in their initial positions.

Then we have defined the states hierarchy of the figure 4. The main structure criteria are :

- Added value (In or out of production)
- Type of control (automatic or manual)
- Energy availability

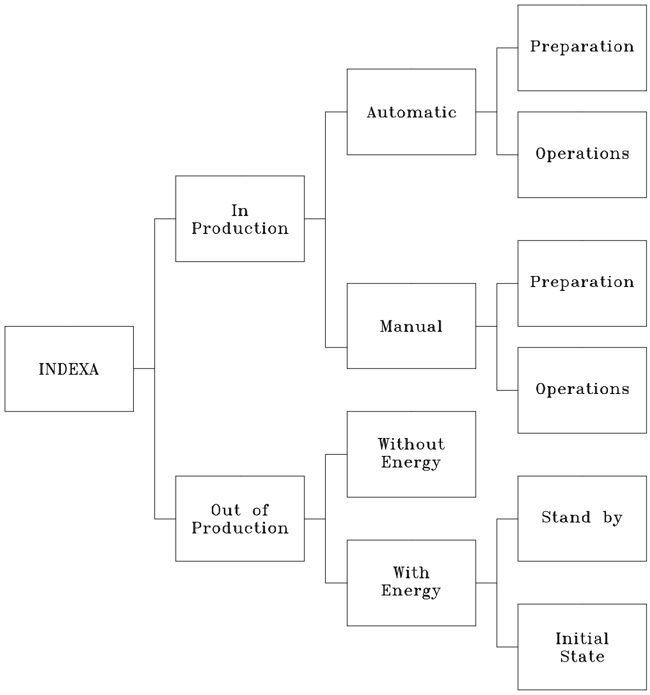


Figure 4) States hierarchy considered

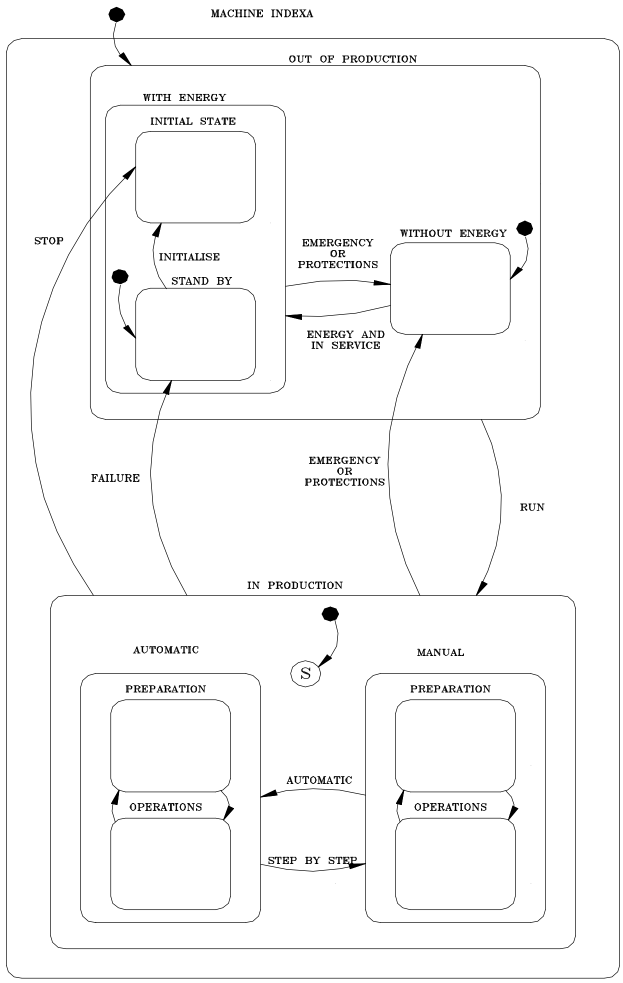


Figure 5) First statecharts model

To create the first sketch of the INDEXA statecharts model, the state hierarchy presented in figure 4 has been used. We obtain the model of the figure 5 in which the state hierarchy is the previous one and the transitions are those authorised with regard to the machine requirements.
This first model does not take into account the actuators technology (e.g. mono-stable or bi-stable devices). In order to generate the physical outputs of the control system, a more detailed model is to be developed. To elaborate this model, the previous states are refined in sub-states just into the lower control level. For instance, the figure 6 shows the refinement of the Operations state included in the Automatic super-state of the Figure 5.

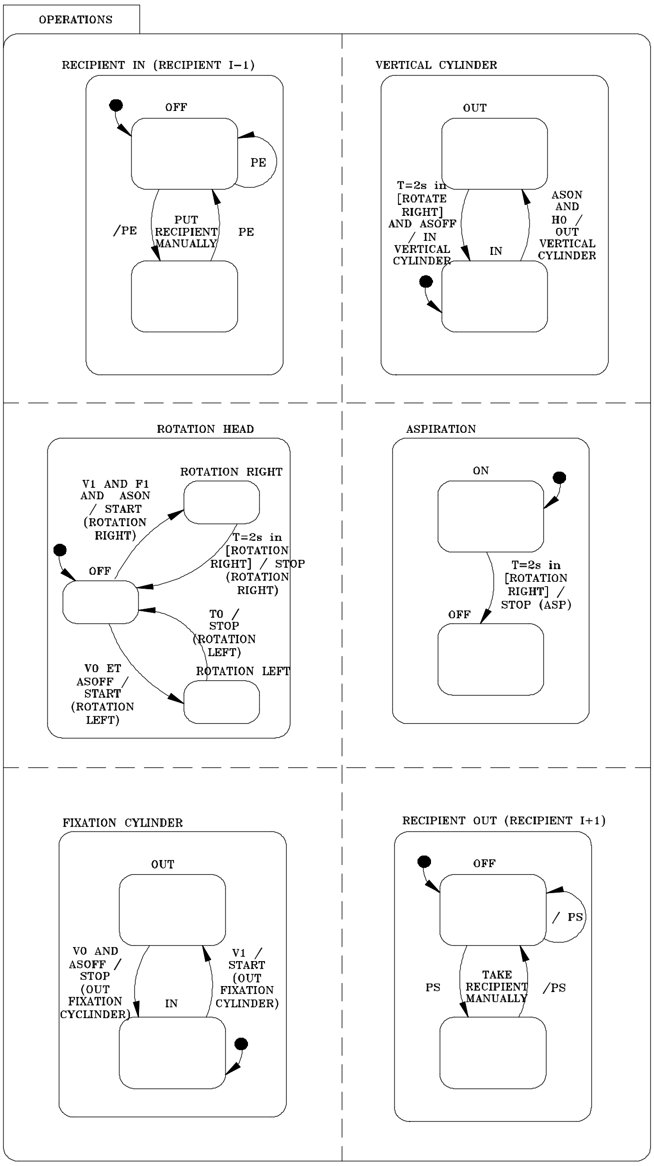


Figure 6) State refinement

During the Operations state, six actuators are to be controlled independently. We have therefore defined six parallel states, each of them describing the control of a given actuator. These six actuators do not have the same behaviour: some are bi-stable devices, others are mono-stable devices. In order to illustrate how the physical behaviour is taken into account in the statecharts model, we are going to detail the Vertical cylinder and Fixation cylinder states.

The vertical cylinder is a bi-stable device. In its control model, two output signals are used (Out vertical cylinder and In vertical cylinder) and we must assume that each of them is TRUE only during the corresponding transition, but not during the states. On the contrary the fixation cylinder is a mono-stable device. Its only control signal (Out fixation cylinder) must be TRUE during the transition from the In state to the Out state and also during the Out state. It is put to FALSE during the reverse transition.

This simple example shows clearly that a statecharts model aiming at entirely describing the control of actuators has to include an explicit description of the behaviour of output signals.

For all the other sub-states the refinement is similar. The size of the global model does not allow us to include it in this paper. Nevertheless we can give some of its relevant characteristics:

- This model is composed of 65 elementary states (states which can not be refined) and 32 macro-states (states including other states)
- A five levels hierarchy has been defined to structure this model.
- Most of the transitions labels includes just a trigger; action are used only in the labels of the transitions linking the elementary states, like in figure 6.

3 – Implementation

The INDEXA machine is controlled by a Programmable Logic Controller (PLC). Instead of the programming environment supplied with this controller, we have decided to use the ISAGRAF tool in order to implement the control model. This software tool enables one to develop programs in IEC 61131–3 standard languages, to simulate them and to generate code for a target controller.

We have only considered in our study the Sequential Function Charts (SFC) and Ladder Diagram (LD) languages, because these languages are the most commonly used for PLC programming in industry and because of the recent advances in the field of formal verification, [Lampérière.99], [Rossi.00], [Lampérière.00]. Moreover, the ISAGRAF software allows one to build a hierarchy of SFC by defining "father" and "son" SFC.

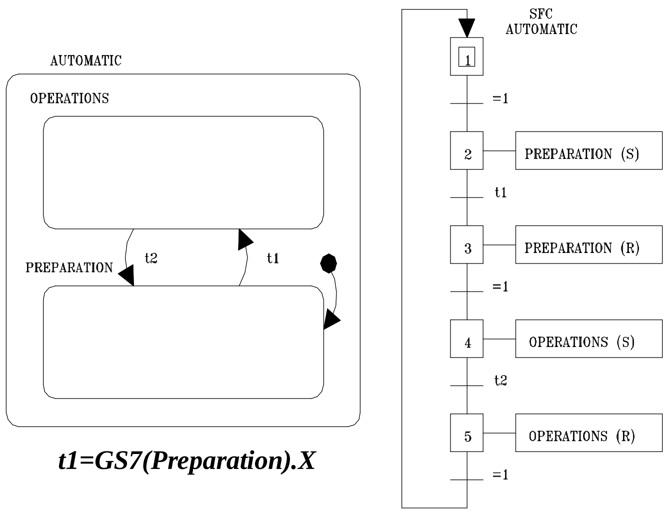
We have elaborated our PLC program from the statecharts model in the following way :

- The SFC program has the same structure as the statecharts model presented in figure 4. For instance, the

SFC “INDEXA” is the main program and is the father of the “In Production” and “Out of Production” SFC and the programs “Automatic” and “Manual” are the sons of the “In Production” program.

- The ladder is used for generating the physical outputs of the system.

The figure 7 shows a statecharts model (Automatic) and its translation into an SFC program.



t1=GS7(Preparation).X

Figure 7) State Automatic and its translation into an SFC program “Automatic”

This figure shows that each macro-state gives rise to two steps, the first one setting the SFC program corresponding to this macro-state and the other resetting it. For instance, when the step 2 of the “Automatic” SFC is activated, the “Preparation” SFC program (see figure 9) is set and can then run. When the condition “t1” becomes true the transition between the steps 2 and 3 is fired, the step 3 is activated and the “Preparation” SFC program is reset (no active step). This condition “t1” is merely “Step 7 of Preparation SFC program is active”, which enables us to synchronise the two SFC programs (“Automatic” and “Preparation”). At the end of the “Preparation” program indeed, the transition between the steps 2 and 3 of the “Automatic” SFC is fired, then the transition between the steps 3 and 4 is also fired which produces the set of the “Operations” SFC in compliance with the statecharts model.

The “Preparation” state presented in figure 8 is the base to the SFC and LD program “Preparation” in figure 9. In this case four sub-states have simultaneous development, so, to the correspondent SFC, there are four parallel and independent evolutions (Figure 9). The transitions are the transitions of the sequences and the system’s physical outputs are represented in the ladder program. These outputs are presented as follows:

- Bi-stable devices: The physical output exists between the states.
- Mono-stable devices: The physical output exists between and during the states.
- ON/OFF systems: The physical output exists only during the states.

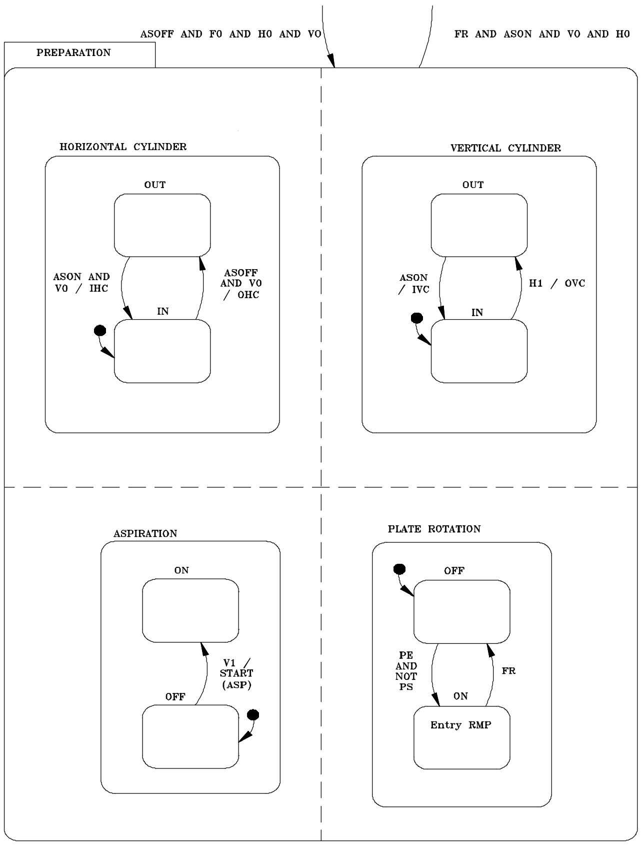


Figure 8) “Preparation” state included in “Automatic” state

The following variables are used in the figures 7, 8 and 9:

H0 – Horizontal cylinder in
H1 – Horizontal cylinder out
V0 – Vertical cylinder in
V1 – Vertical cylinder out
ASON – Aspiration ON
ASOFF – Aspiration OFF
PE – Piece into the place to go in
PS – Piece into the place to go out
OHC – Out horizontal cylinder
IHC – In horizontal cylinder
OVC – Out vertical cylinder
IVC – In vertical cylinder
ASPIRATION – Aspiration ON
PMR – Table motor rotation

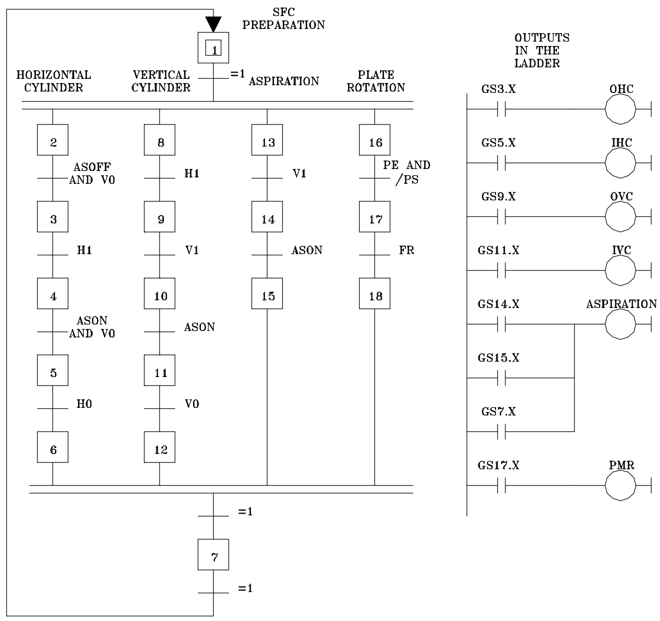


Figure 9) “Preparation” SFC. Program-son of “Automatic”

All the statecharts states have been translated into the correspondent SFC programs according to the structure presented in figure 4, and all the physical system outputs have been represented in the Ladder program, following the presented rules.

The PLC program has been tested by using simulation tools of the ISAGRAF environment. The obtained behaviour is the expected one.

4 – Conclusions and perspectives

This study has shown that the statecharts formalism can be used to elaborate a production system control model providing that :

- A state hierarchy based on operating modes of the system has been previously clearly defined
- The relations between the internal evolutions of the model and those of its outputs have been specified.

We think that statecharts have been mainly developed for software control. The use of this formalism for physical devices control modelling requires these two adaptations.
Our work has also permitted us to systematise the elaboration of a PLC program in SFC and LD languages from a statecharts model.

Future works are numerous and significant :

- First the definition of a sound temporal model of statecharts enabling to link internal evolutions to input/output changes is to be proposed
- Properties checking on the control model and on the PLC program should be performed in order to ensure the safety of the control.
- Extension of the developed method to other PLC programming languages: Structured Text (ST), Instruction List (IL) and Function Block Diagram (FBD).

List of References

[Duméry.94] : J.J. DUMERY. *Modélisation des modes de marche des systèmes automatisés de production.* Mémoire de recherche, DEA de production automatisée, Ecole Normale Supérieure de Cachan. Juillet 1994.

[Guéguen.96] : H. GUEGUEN. *Synchronisme dans l’évolution des statecharts.* Actes Congrès AFCET MSR'96 : Modélisation des systèmes réactifs. Brest, France. pp. 285-292. 28-29 mars 1996.

[Harel.87] : D. HAREL. *Statecharts : a visual formalism for complex systems.* Science of computer programming North Holland. Vol. 8 pp 231-274. 1987.

[Harel.88] : D. HAREL. *On visual formalism* . Communications of the Association Computer Machine (ACM). Vol. 31 N°5 pp 514-530, mai 1988.

[Harel.96] : D. Harel, A. Naamad, *The STATEMATE Semantics of Statecharts*, Wisdom Technical reports in Computer Science, CS95-31, 1995, revised in july 1996, ACM TOSEM, 1996.

[IEC 61131-3] : Programmable Controllers – Programming languages, march 1993.

[IEC 60848] : Preparation of function charts for control systems, 1988.

[ISAGRAF] : ISAGRAF MANUAL . *CJ International.*

[Lampérière.00] : S. Lampérière-Couffin, J.-J. Lesage*: Formal verification of the sequential part of PLC programs*, Wodes2000 - 5th Workshop on Discrete Event Systems, Ghent (Belgium), 21 August to 23 August 2000 pp. 247-254.

[Lampérière.99] : S. Lampérière-Couffin, J.-J. Lesage , J.-M. Roussel *: Formal validation of PLC programs : a survey*, IFAC - IEEE European Control Conference 1999 (ECC’99), paper n°741, Karlsruhe (Germany), 31. August - 3. September 1999.

[Rossi.00] : O. Rossi, O. De Smet, S. Couffin, J.-J. Lesage, H. Papini, H. Guennec *: Formal verification : a tool to improve the safety of control systems*, SafeProcess’2000; pp. 885-890, IFAC Budapest, 14-16 june 2000.