Improving the Teaching of Discrete-Event Control Systems Using a LEGO Manufacturing Prototype

Arturo Sanchez and Jorge Bucio

***Abstract*—This paper discusses the usefulness of employing LEGO as a teaching-learning aid in a post-graduate-level first course on the control of discrete-event systems (DESs). The final assignment of the course is presented, which asks students to design and implement a modular hierarchical discrete-event supervisor for the coordination layer of a complete automated manufacturing system (AMS) built using LEGO blocks. A design approach frequently used in manufacturing is introduced to unify design criteria and nomenclature. Software tools are provided for all calculation and translation tasks required for the implementation of the supervisor architecture in a programmable logic controller (PLC). The assignment reinforces all the basic concepts of supervisory control theory taught during the course. It provides hands-on experience of the advantages and disadvantages and of the important theoretical and practical issues related to the use of DES controllers in an engineering application.**

***Index Terms*—Automated manufacturing systems (AMSs), coordination control, discrete-event systems (DESs), supervisory control theory (SCT).**

I. Introduction

THE SUBJECT of discrete-event systems (DESs) has been established over the past 15 years as being highly relevant for the control systems community (both academic and industrial). The study of control-related issues, both theoretical and practical, is enriched with other approaches from operations research, software engineering, and computing. This allows a wider exploration of the topic to include, for example, the software implementation of formally-correct-by-construction controllers that would have helped to avoid the occurrence of the fatal accident in the isomerization unit at the Texas City refinery, Texas City, TX, in 2005 [1].

DES control has been taught at the Advance Engineering Unit of Cinvestav (Cinvestav-Gdl), Mexico, since 1995 as part of the Master's program in Control. Two 60-h courses are offered to groups of between 10 and 14 students. The first is part of the compulsory curriculum and presents an introduction to DES control issues from a systems perspective, based on supervisory control theory (SCT). The second is an elective and is usually employed to explore advanced issues on DES. The first course follows the syllabus of Wonham's monograph on SCT [2], covering the material from Ch. 3, with basics on finite-state automata and regular languages, up to Ch. 5 which deals with the existence of modular-hierarchical supervisory control architectures. Class exercises and assignments include both theory and calculation. Application examples and exercises are taken mainly from manufacturing systems, which is one of the application domains at Cinvestav-Gdl. Publicly available software tools (TCT [2], Supremica [3], SSPC [4]) are employed for the required calculations. In order to reinforce and clarify the basic theoretical concepts, a LEGO implementation of Example 3.7.1 from Wonham's monograph was carried out during the 2005 and 2006 courses, using Visual Basic as the programming platform [5]. This exercise helped some students to assimilate the basic concepts of DES and SCT. The enthusiasm among the students for using a familiar toy as a scientific modeling tool was noticeable. Surveys of the students indicated that more than half had previous exposure to LEGO, and more than 80% had previous experience of designing or using logic controllers, either with programmable logic controllers (PLCs) or microcontrollers. However, none of the students had designed logic controllers for manufacturing systems using engineering-based approaches. Based on the previous positive results and the student profile, a project assignment was devised to address the rest of the material covered in the course. The project, assigned to student pairs, consisted in the design and implementation of a resource-coordination controller for a LEGO maquette of an automated manufacturing system (AMS). The "products" were fabricated by the sequential "assembly" of LEGO blocks. The control architecture was deployed in a PLC, which is a type of control device frequently used in manufacturing systems. The project was set as the final assignment of the course and was first piloted during the academic year 2008. Improved during the 2009 course, the description presented here is the assignment from the 2010 course.

The rest of this paper is organized as follows. Section II describes the project assignment and the activities involved. In Section III, the LEGO AMS is introduced with a description of each of its main elements. As an example of the hierarchical architectures for resource-coordination control resulting from the exercise, a three-level modular architecture is described that has frequently been proposed by the students and that has been used previously for research purposes [6]. SCT-supervisors are synthesized for each resulting module. An example is presented following the synthesis procedure employed by the students. The PLC implementation is briefly described in Section VI. The target architecture is a standard industrial PLC and its programming software (a Siemens Simatic S7-300). Students

Manuscript received February 24, 2011; revised June 24, 2011; accepted October 10, 2011. Date of publication November 08, 2011; date of current version July 31, 2012. The work of J. Bucio was supported in part by CONACYT, Mexico, under Scholarship 234880.

The authors are with the Centro de Investigacion y de Estudios Avanzados (Cinvestav), Unidad de Ingenieria Avanzada, Zapopan 45091, Mexico (e-mail: arturo@gdl.cinvestav.mx).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2011.2173688

are exposed to two common implementation languages: structured-text (ST) for SCT-supervisors and ladder diagrams for the input–output (I/O) layer of DES models for sensors and actuators. Finally, Section VII comments on the students' response to this assignment, the observed benefits, and future improvements to the assignment.

II. Project Assignment

The assignment objective is to design and implement in a PLC a modular hierarchical discrete-event controller, termed a *coordination controller*, for an AMS maquette built with LEGO blocks. These controllers supervise resource-allocation tasks of equipment groupings during product manufacturing. Therefore, DES-supervisory controllers are suitable for supervising equipment start–stop events established by production schedules. The control architecture must be designed and implemented in one calendar week, including the writing of the corresponding report.

The project assignment is introduced in a 2-h session. First, the LEGO AMS is presented and the assignment objective is explained by the instructor. The students must design a modular-hierarchical coordination architecture capable of supervising the execution of any production schedule (e.g., the fabrication of a number of assemblies using specific sequences of LEGO blocks and AMS resources) within the physical limitations of the AMS (e.g., maximum number of blocks at any time in a working station). In order to standardize nomenclature and design strategies, the instructor introduces the ISA-S88 standards for industrial batch control [7] together with examples of their use. The use of these standards exposes the students to an engineering approach and its associated nomenclature that is well accepted in manufacturing and process industries. For the synthesis of the individual supervisors, students are instructed to follow the procedure proposed in [4], which was introduced as part of the course syllabus during Ch. 3 and was a compulsory procedure for homework assignments. Therefore, since students are already trained in the use of available calculation tools, they can focus on solving the design problem. The behavior specifications must be modular, and it is not acceptable that one specification should capture the whole desired behavior of a module. Two calendar days later, in a second 2-h session, each team delivers a presentation in which the strategy adopted for designing the coordination control architecture is explained, together with its proposed architecture and corresponding synthesized supervisors. Engineering and theoretical issues are discussed: For instance, the architecture may restrict the efficient use of resources, or behavior specifications may give rise to a too stringent *minimally restrictive* closed-loop behavior.

Once all the architectures are reviewed, the instructor introduces the use of the PLC and its programming tools, together with the translation tools available for this task. After this session, the students must carry out any accepted modification to the architecture or supervisors suggested by the instructor or other students and proceed to translate and implement the code in the PLC. The instructor supervises these activities. This usually takes around three to four days because only one AMS maquette is available, so students must work in shifts. Five calendar days after the review session, each team must deliver a report on the assignment and show the instructor how the proposed control architecture deals with the "production programs" in the AMS maquette. The instructor provides, on the spot, a production program to the team (e.g., fabricate two assemblies of one red block and one black block at workstation 1 and three assemblies of two red blocks in workstation 2). The team must implement in the PLC the given production program and show the proper operation of the AMS. The instructor grades the presentation, the proposed architecture, its implementation in the PLC, and the report.

III. LEGO AMS

The LEGO AMS maquette is based on a laboratory prototype. The maquette was designed as a low-cost, easy-to-carry test-bed for teaching purposes [8], [9] as well as for research. Its manufacturing functionality is intentionally kept simple. Nevertheless, it is complex enough to make it extremely difficult and very time-consuming for the development of a DES control system without using systematic techniques and automated calculation tools. Thus, the assignment makes clear not just the advantages but also the necessity of employing formal (e.g., synthesis) techniques to guarantee the correct development of DES controllers satisfying specific behavior (i.e., safety and operational) specifications, even for systems with a certain degree of complexity.

LEGO has been widely used for teaching and research projects in a range of disciplines, demonstrating its versatility and affordability (e.g., [10], [11]). In DES-related projects, LEGO has also been employed for research purposes [6], [12]. The authors are aware of another effort using LEGO-based maquettes that has not been documented in the open literature. The LEGO devices are sufficiently accurate and the sensors sufficiently sensitive to carry out the tasks simulated by the AMS, compared to other available alternatives for building prototypes at the same scale. The signal conditioning issues encountered [10] when interfacing the LEGO devices with a PC or a PLC (e.g., working cycles for sensors, number of I/O signals) were resolved using simple electronics [13], [14]. Other favorable aspects were the small initial investment and low maintenance costs. The total cost of the required components for this 40×40 cm^2 maquette was less than US $800. The components employed in the construction of the electronic communication interface with the PLC, including the PCB, amounted to less than US $100 (circa 2009).

A schematic plant-view of the original laboratory AMS is shown in Fig. 1. Two dispatchers (D1 and D2) deliver raw-material units A and B to two workstations (WS1 and WS2) using a rectangular conveyor belt. Each workstation is composed of a feeder, a processing machine, and an unloading device for finished parts. Processing tasks are executed to manufacture products according to given ISA-S88 *master recipes procedures* (i.e., production programs).

The AMS maquette was built with standard LEGO blocks as shown in Fig. 2. The design was carried out using computer-aided design (CAD) tools frequently employed in the LEGO community [8], [9]. Raw-material units are simulated by individual 2×2 blocks. The dispatcher mechanism is shown in

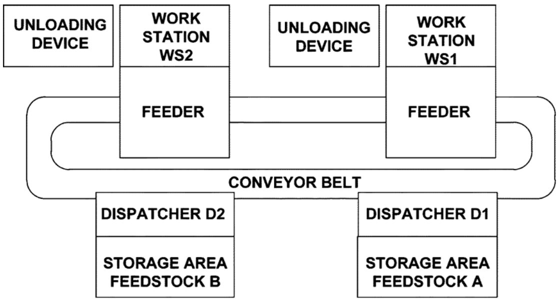

Fig. 1. Block diagram of AMS prototype.

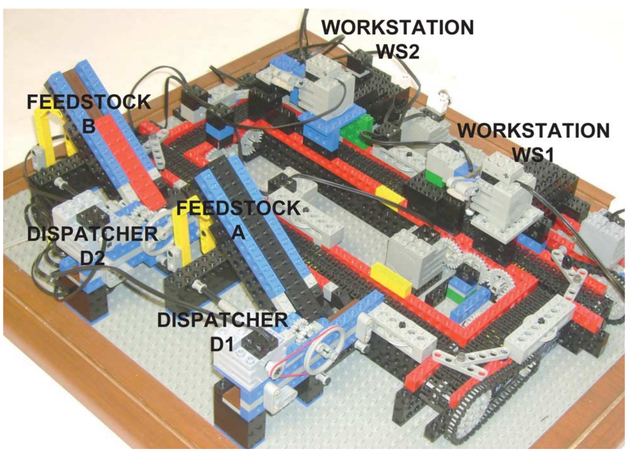

Fig. 2. AMS implementation with LEGO.

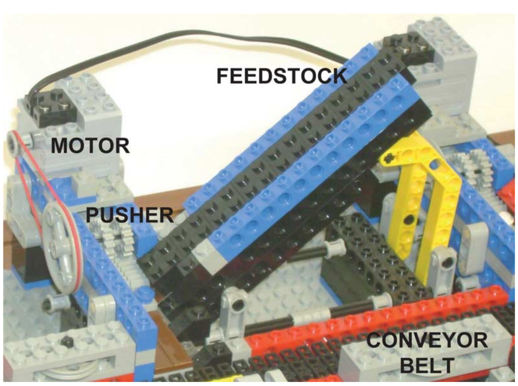

Fig. 3. Detail of feedstock dispatcher.

Fig. 3. A retracting pusher, powered by a motor, pulls the incoming block into the conveyor belt and then returns to its original position. Details of the workstation are shown in Fig. 4. The feeder assembly is placed above the conveyor belt. It contains a light sensor, a crankshaft powered by a motor, and a touch sensor. The light sensor detects the raw-material block of the desired color. Once the block is detected, the crankshaft pushes the block into the processing machine. The touch sensor detects the turns of the crankshaft. The processing machine is a receptacle from where the block is (or blocks are) removed once the processing time expires, using a similar mechanism to the dispatcher pusher.

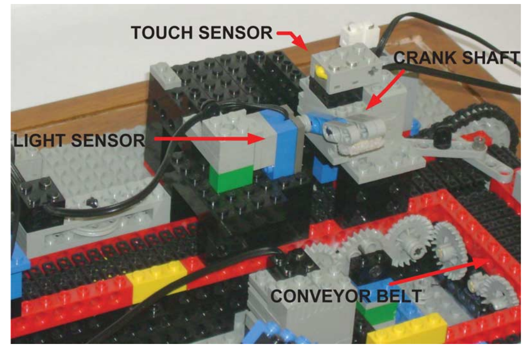

Fig. 4. Detail of workstation.

TABLE I
ISA-S88 EQUIPMENT MODEL OF THE AMS MAQUETTE

CELL	EQ. UNIT	EQ. MODULE	ELEMENTARY EQ. COMPONENTS
AMS	DRM	Dispatchers	Feeding motors
		Conveyor belt	Unit counter
	WS1	WS1 feeder	Motor, light and touch sensors
		WS1 machine	Delivering motor
	WS2	WS2 feeder	Motor, light and touch sensors
		WS2 machine	Delivering motor

TABLE II
ISA-S88 PROCEDURAL CONTROL MODEL OF THE AMS MAQUETTE

PROC.	UNIT PROC.	EQUIPMENT PHASE
Manufacture Product	Dispatch raw material (UPDRM)	Dispatch raw material A or B (PhDRMAB)
	Process in WS1 (UPPWS1).	Feed raw material A or B (PhFRMAB)
		Process and deliver (PhPD)
	Process in WS2 (UPPWS2).	Feed raw material A or B (PhFRMAB)
		Process and deliver (PhPD)

IV. COORDINATION CONTROL ARCHITECTURE

The ISA-S88 standards for batch control [7] are extensively used in the manufacturing and process industries. They provide guidelines for the design and implementation of a class of control systems, taking into consideration the particularities of these industries. Central to the standards are the notions of *process*, *procedural control*, and *equipment* models. The process model describes the process-oriented tasks that can be carried out in a given facility by executing procedural activities dictated by the procedural control model. The procedural activities are related to equipment modules established in the equipment model. Well-structured hierarchical models and terminology are provided by the standards. In this context, coordination control deals with the resource allocation required to carry out a production schedule. Tables I and II show the ISA-S88 equipment and procedural control models for a coordination control layer of an AMS LEGO maquette frequently proposed by students as a solution in which the supervisory models at each level are *trivially nonconflicting*. Therefore, from an engineering point of view, it can be considered a good solution because the *phase* and *unit-procedure* modules at each level do not share resources. This may favor a safer operation and an easier-to-maintain control system.

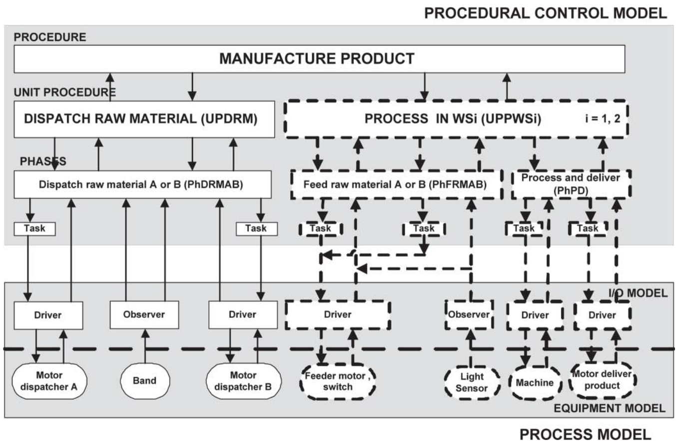

Fig. 5. Hybrid model of AMS coordination control layer.

Both phase and unit procedure control modules deal with equipment operation of the AMS. The procedure level contains the event (i.e., fabrication) sequence [15] for manufacturing specific products, modeled as an ordered set of events changing the phase-status. These events are employed as the linking mechanism between the various levels. The event sequence may be the result of a previous planning stage. In this exercise, the production program is given to the students once they have demonstrated to the instructor the operation of their coordination architecture for the AMS. They must then translate this into an FSM capturing the adequate high-level event sequences to properly drive the execution of the equipment unit for the fabrication of the required products. The students must make decisions on resource allocation or fabrication strategies that, in a manufacturing system, are usually part of the upper production programming layer. The students enter this information in the procedure level of the hierarchical architecture implemented in the PLC.

In the proposed architecture, workstations WS1 and WS2 are considered *equipment units*, as are both raw material dispatchers and the conveyor belt, identified with the label DRM. Equipment phases are associated with equipment modules. The resulting equipment modules and elementary components (both motors and sensors) are listed in their corresponding columns of Table I. A graphical representation of the procedural control model of the coordination layer is shown in Fig. 5. For reasons of space, broken lines are used to show that modules are repeated for each workstation. An I/O layer is introduced to communicate the equipment phases with manufacturing equipment using *device drivers* and *state observers* [16]. A device driver translates control instructions from the equipment phase to the equipment, and vice versa, calculating the state of its associated manufacturing equipment using a discrete-event model if required. The state observers extract occurrences of discrete events from continuous measurements. This I/O layer has been implemented in the PLC, and students are not allowed to modify it. Students are given DES models and the I/O nomenclature, and must then decide what information to use to construct the control architecture and how to pair the PLC inputs and outputs with uncontrollable and controllable transitions, as shown in Section V. The implementations of discrete-event models and state observers are briefly discussed in Section VI.

V. Supervisor Synthesis

Supervisors are paired to equipment phases and unit procedures according to the coordination layer model of Fig. 5. Both phase and unit procedure supervisors must enforce safety and functional specifications during operation of the AMS, leaving to the production programs the definition of the phase sequencing required for manufacturing specific parts.

Phase supervisors deal with the associated components of the equipment unit, while unit procedure supervisors ensure the safe execution of phases as requested by the production program. In the architecture presented here, three different supervisors were synthesized for the phase level: "Dispatch raw material A or B (PhDRMAB)," "Feed raw material A or B (PhFRMAB)," and "Process and deliver (PhPD)." In order to give an idea of the complexity of the synthesis tasks, Fig. 6 shows the input and resulting automata for the "Feed Raw Material A or B (PhFRMAB)" phase. The figure includes the elementary component models (process equipment and phase status information) provided to the students by the instructor, the behavior specifications proposed by a team, and the automaton representation of the closed-loop behavior FSM acting as a supervisor. Table III lists the events along with their physical interpretation. The plant model is the result of the synchronous product of both the elementary components and the causal behavior automaton indicating that a phase cannot finish before the motors are switched off. Five behavior specification automata describe the desired operation. For instance, specification "Motor start-up" declares that the motor must be switched on only when raw material is detected. The rest of the supervisors are synthesized in the same fashion. Automata sizes for all synthesized supervisors are included in Table IV. As can be seen, state spaces are rather small in all cases. Verifying structural properties can therefore easily be carried out by inspection.

VI. PLC Implementation

Each synthesized supervisor is realized as an FSM $S = (Y, \Sigma, \delta_S, y_0, Y_m)$ [2] and translated into Structured Text (ST) language by mapping the transition function $\delta_S : Y \times \Sigma \rightarrow Y$ into a CASE structure of ST. States are modeled as integer-type variables. Controllable and uncontrollable transitions are modeled respectively by Boolean-type output variables (i.e., actions to be executed from a given state) and input variables (i.e., input signals required by the controller at a particular state). For example, if an input variable is true, a change in state is exerted (e.g., IF $\mathrm{T35} = \mathrm{TRUE\ THEN\ STATE} := 1\ \mathrm{END\_IF}$). If a controllable transition is enabled in the state, its corresponding Boolean variable is set to TRUE and variable STATE is assigned the value established by δ_S. As an example, Fig. 7 shows the corresponding ST translation of the declaration section and the first four states of the "Feed Raw Material A or B in WS1" supervisor.

DES models and state-observers in the I/O layer are translated into ladder logic language following the same principle as above, but with states and transitions modeled as *contacts* whose value is changed by SET and RESET coils. As an example, Fig. 8 shows the ladder logic diagram (LD) of an FSM

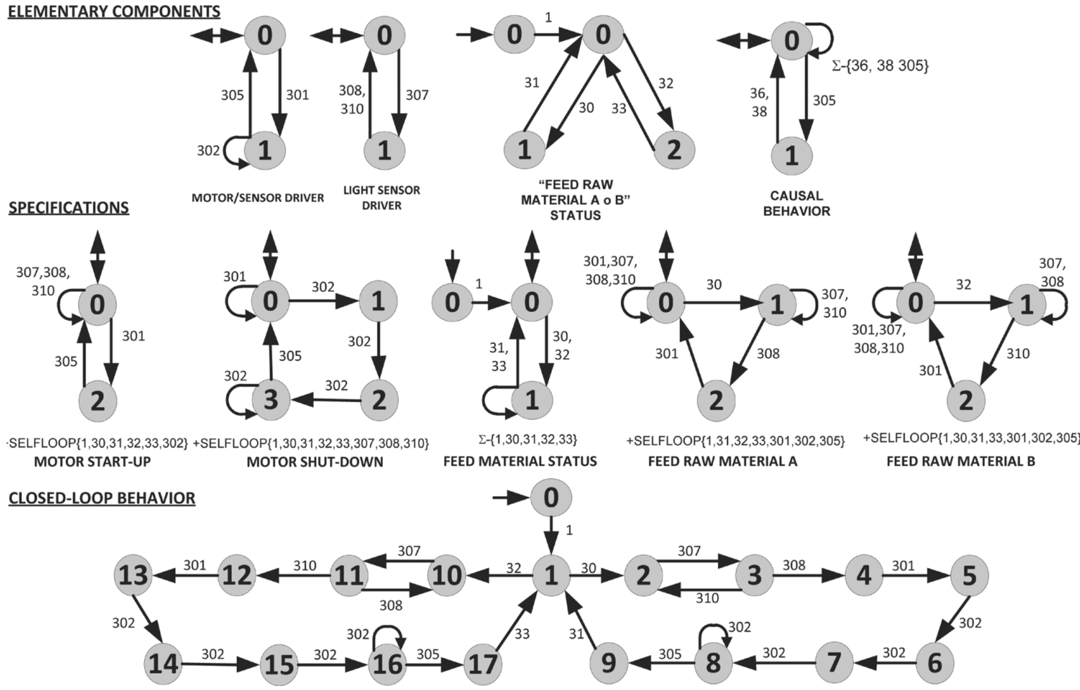

Fig. 6. Elementary components, specifications, and resulting supervisor of "Feed Raw Material A or B in WS1" phase.

TABLE III
EVENTS FOR "FEED RAW MATERIAL A OR B IN WS1" PHASE

Evt	Meaning	Type
1	Phase "FRMAB" start-up	Controllable
30	Phase "FRMA" start-up with raw material A	Uncontrollable
31	Phase "FRMA" finishing with raw material A	Controllable
32	Phase "FRMB" start-up with raw material B	Uncontrollable
33	Phase "FRMB" finishing with raw material B	Controllable
50	Phase "Op. WS" finishing	Uncontrollable
51	Phase "Op. WS" start-up	Controllable
301	Switch-on motor left-wise	Controllable
302	Touch sensor being activated	Uncontrollable
303	Switch-on motor right-wise	Controllable
305	Switch-off motor	Controllable
307	Switch-on light sensor	Controllable
308	Light sensor detects RM A	Uncontrollable
310	Light sensor detects RM B	Uncontrollable

TABLE IV
SIZES OF THE AMS SUPERVISORY MODULES

Phase Supervisor	# States		
	Plant	Specs	Sup
Dispatch raw material (A and B)	640/748	120	23
Feed raw material (A and B)	16/18	80/112	18
Process and deliver	40	36	12
Unit Procedure Supervisor	**# States**		
	Plant	**Specs**	**Sup**
Dispatch raw material	4	-	4
Process in WS i	6	6	8

modeling a feeding motor. The first rung models the execution of the events activating the motor in the desired direction. The second rung corresponds to the event of stopping the feeding motor; the SET coil "FM1_S0" restores the state-contact to its initial value. These translation procedures are implemented in a software application provided to students [14].

```
FUNCTION_BLOCK FC28
  VAR_INPUT
    T30 : BOOL;
    T32 : BOOL;
    T308 : BOOL;
    T310 : BOOL;
  END_VAR
  VAR_OUTPUT
    T1   : BOOL;
    T307 : BOOL;
  END_VAR
  VAR
    state : INT := 0;
  END_VAR
  BEGIN
  CASE state OF
    0:
      T1 := TRUE;
      state := 1;
    1:
      IF T30 = TRUE THEN
        T30 := FALSE;
        state := 2;
      END_IF;
      IF T32 = TRUE THEN
        T32 := FALSE;
        state := 10;
      END_IF;
    2:
      T307 := TRUE;
      state := 3;
    3:
      IF T308 = TRUE THEN
        T308 := FALSE;
        state := 4;
      END_IF;
      IF T310 = TRUE THEN
        T310 := FALSE;
        state := 2;
      END_IF;
  ELSE
    state := 0;
  END_CASE
END_FUNCTION_BLOCK
```

Fig. 7. ST implementation for the "Feed Raw Material A or B in WS1" supervisor.

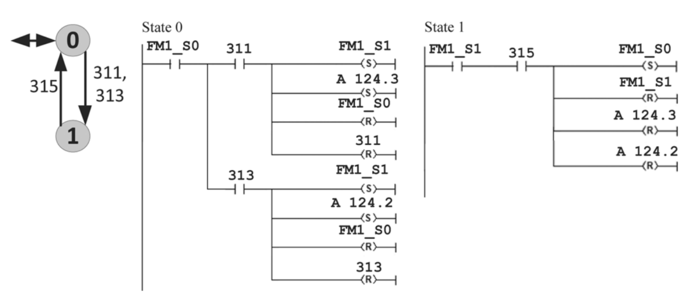

Fig. 8. LD of a motor FSM model.

VII. Conclusion

This exercise provides the students with an industrial-type case study covering most of the concepts introduced in this first course in DES control. It clearly illustrates some of the advantages of using a formal synthesis approach (e.g., correct-by-construction controllers), disadvantages (e.g., difficult to handle functional specifications, or explicit representation of the controller FSM giving rise to very large and difficult to maintain PLC-functions) and other important theoretical issues to take into consideration when doing an implementation (e.g., maximum permissiveness of controllers). Surveys conducted of the 2009 and 2010 students showed that they considered the assignment to be a useful aid for clarifying and reinforcing the basic DES concepts on SCT. The use of industrial standards and tools for designing and implementing the controllers were also welcomed. The students' pleasure in using a familiar toy as a learning vehicle and a scientific tool was noticeable. Due to the intensive use of the AMS during the assignment, it is advisable to have on hand spare parts such as motors, touch sensors, and cogwheels. Since most students have already been trained in the use of logic controllers before taking this course, a comparative exercise is planned for the next academic year. The assignment will be introduced at the beginning of the course, and the students will be requested to design and implement the DES controllers using any previous knowledge they may have. It is expected that at the end of this first part of the course, the students will have a sound basis into which to incorporate new concepts introduced as the course progresses. The exercise will be repeated at the end of the course in the terms presented here. The students will then be able to appreciate the advantages of a systematic approach based on engineering design tools reinforced with formal synthesis methods.

Acknowledgment

A. Sanchez gratefully acknowledges the M.Sc. students of the 2008, 2009, and 2010 courses who enthusiastically participated in testing the equipment and the assignment, as well as contributing suggestions to improve them. D. Camacho, M.A., provided the framework to construct the teaching–learning case. P. King, M.Sc., read an early version of this manuscript and suggested the presentation approach. Feedback from the reviewers is also deeply appreciated.

References

[1] D. Holmstrom, "Chemical Safety Board. BP investigation," U.S. Chemical Safety Board, Washington, DC, 2005 [Online]. Available: http://www.csb.gov

[2] W. M. Wonham, "Supervisory control of discrete event systems," 2008 [Online]. Available: http://www.control.utoronto.ca/people/profs/wonham

[3] K. Akesson, M. Fabian, H. Flordal, and R. Malik, "Supremica, An integrated environment for verification, synthesis and simulation of discrete event systems," in *Proc. 8th WODES*, Ann Arbor, MI, 2006, pp. 384–385.

[4] A. Sanchez, J. Douriet, and E. Ramirez, "Formal synthesis of a class of discrete-event controllers for large manufacturing systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 662–669, Jul. 2007.

[5] E. Hernandez, E. Aranda-Bricaire, and A. Sanchez, "Hands-on exercises of supervisory control of discrete event manufacturing systems," presented at the XIV Cong. Latinoamer. Control Autom., Santiago de Chile, Chile, Aug. 24–27, 2010.

[6] A. Sanchez, E. Aranda-Bricaire, F. Jaimes, E. Hernandez, and A. Nava, "Synthesis of product-driven coordination controllers for a class of discrete-event manufacturing systems," *Robot. Comput.-Integr. Manuf.*, vol. 26, pp. 361–369, 2010.

[7] "Batch control systems. Part 1. Models and terminology. Standards," ISA, Research Triangle Park, NC, ISA-88.01, 1995.

[8] E. Hernandez, "Hierarchical-modular modeling and control of resource coordination for flexible manufacturing systems," (in Spanish) Master's thesis, Seccion de Mecatronica, Cinvestav, Mexico City, Mexico, 2005.

[9] J. A. Nava, "A modelling framework for the design of industrial coordination control architectures," (in Spanish) Master's thesis, Cinvestav, Guadalajara, Mexico, 2006.

[10] P. Gawthrop, E. W. , and McGookin, "Using LEGO in control education," in *Proc. 7th IFAC Symp. Adv. Control Educ.*, Madrid, Spain, Jun. 21–23, 2006, pp. 1230–1235.

[11] D. Benedettelli, M. Casini, A. Garulli, A. Giannitrapani, and A. Vicino, "A LEGO Mindstorms experimental setup for multi-agent systems," in *Proc. ISIC*, Jul. 2009, pp. 1230–1235.

[12] V. Chandra, Z. Huang, and R. Kumar, "Automated control synthesis for an assembly line using discrete event system control theory," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 33, no. 2, pp. 284–289, May 2003.

[13] F. A. Garcia, "Formal development of logic controllers. A case study in automated manufacturing systems," (in Spanish) Master's thesis, Unidad de Ingenieria Avanzada, Cinvestav, Guadalajara, Mexico, 2007.

[14] J. Bucio, "Formal implementation of modular-hierarchical discrete-event supervisors for automated manufacturing systems," (in Spanish) Master's thesis, Cinvestav, Guadalajara, Mexico, 2011.

[15] "Enterprise—Control system integration. Part 1: Models and terminology. Standards," Instrument Society of America, Research Triangle Park, NC, Tech. Rep., 1999.

[16] A. Sanchez, L. F. Parra, and R. B. S. Macchietto, "Hybrid modelling and dynamic simulation of automated batch plants," *ISA Trans.*, vol. 42, pp. 401–420, 2003.

Arturo Sanchez received the B.Sc. degree in chemical engineering from the Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), Tlaquepaque, Mexico, in 1985, the M.I. degree in chemical engineering from Universidad Autónoma Metropolitana (UAM-I), Mexico City, Mexico, in 1989, and the Ph.D. degree in process automation from Imperial College, London, U.K., in 1994.

He is currently a Senior Research Scientist with the Automatic Control Group, Cinvestav-Gdl, Zapopan, Mexico. He has been a Visiting Research Scientist with the Department of Computing, Imperial College; the Group of Applied Mathematics, Mexican Petroleum Institute, Mexico City, Mexico; and the Department of Engineering Sciences, University of Oxford, Oxford, U.K. He has published more than 100 papers and a book and has graduated more than 30 M.Sc. and Ph.D. students. His research interests include the formal development of automation systems and biofuels process engineering.

Jorge Bucio received the B.S. degree in electronics and communications engineering from ITESM University, Guadalajara, Mexico, in 2008, and the M.Sc. degree in electrical engineering from CINVESTAV, Guadalajara, Mexico, in 2011.

His current research interests include discrete-event systems, hierarchical controllers, and the control of manufacturing systems.