Korean J. Chem. Eng., **29**(5), 583-594 (2012)
DOI: 10.1007/s11814-011-0227-x

An automata-based approach to synthesize untimed operating procedures in batch chemical processes

Ming-Li Yeh and Chuei-Tin Chang†

Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, R.O.C.
(*Received 9 May 2011 • accepted 29 August 2011*)

Abstract−Systematic synthesis of untimed operating procedures has always been considered as an important design issue for batch chemical processes. An automaton-based method is developed in the present study to perform this task automatically. On the basis of the proposed methodical model-building principles, two distinct types of automata can be constructed to characterize the plant behaviors and control specifications, respectively. An admissible supervisor can be produced by applying the parallel composition operation with these models. For the purpose of identifying the most efficient operation procedures, the supervisor can then be integrated with a set of auxiliary automata to set the operation target(s) and, also, to impose upper limits on the total numbers of actuator actions and operation steps. Three examples are presented to demonstrate the feasibility and correctness of the proposed approach.

Key words: Automaton, Batch Process, Discrete-event System, Supervisory Controller, Synthesis

INTRODUCTION

Any batch process can be fully characterized with a piping and instrumentation diagram (P&ID) and a sequential function chart (SFC). All hardware items and their interconnections are depicted in the P&ID, while the operation steps and their activation conditions are incorporated in an SFC. Traditionally, the SFCs in batch processes are synthesized manually on an ad hoc basis, and this labor-intensive task often becomes unmanageable as the degree of system complexity increases. To avoid human errors and to ensure operational safety, it is the intention of this research to develop a systematic method to automatically generate the batch operating procedures. Also, for the sake of facilitating clear description of the proposed method, the focus of subsequent discussions will be placed only upon the *untimed* operations, i.e., the batch procedures in which none of the activation conditions involve time measurement.

The original procedure synthesis problem was first defined by Rivas and Rudd [1] in 1974. Extensive studies concerning the design and verification of procedural controllers were carried out in the later years [2-9]. Notice that that this research issue has been addressed on the basis of numerous different modeling/reasoning mechanisms, e.g., the mathematical programming models [10-12], the symbolic model verifiers [13], the AI-based linear and nonlinear planning strategies [14-16], and other qualitative models such as the state graphs [17-19] and Petri nets [20-24], etc. Although interesting results have been obtained in the aforementioned studies, the available methods are still not mature enough for realistic applications in the batch chemical plants.

The supervisory control theory was pioneered by Ramadge and Wonham [25,26]. In this control paradigm, a discrete-event system is characterized with a set of event sequences (or the so-called "language") which can be predicted with a finite-state machine, i.e., automaton. A system supervisor can be synthesized with two distinct automata: the plant model and the specification model. The former is used to represent what a system can do physically, while the latter for defining the "legal" events or actions. Although extensive studies have already been carried out with this modeling approach [27-32], none of them provide a concrete step-by-step strategy to synthesize the optimal operating procedure for any given batch chemical process. It is therefore the goal of this work to develop a systematic method to perform this task automatically.

The proposed implementation procedure consists of four steps. In the first step, automata are constructed to model all components specified in the piping and instrumentation diagram (P&ID) of an *uncontrolled* batch process. The second step is to stipulate the control specifications and build the corresponding automaton models. An *admissible* supervisor can then be automatically assembled in the third step by applying the standard *parallel composition* operation [33] with these automaton models. Finally, for the purpose of identifying the most efficient operating procedure(s), the admissible supervisor is augmented with a set of auxiliary automata so as to set the operation target(s) and to impose an upper bound of the total number of actuator actions and/or operation steps. This final step is accomplished by producing the *supremal controllable sublanguage* [33] on the basis of the aforementioned automata.

The remainder of this paper is organized as follows. To facilitate explanation of the proposed automata-building methodology, the general framework of automaton models and the hierarchical structure of batch processes are first presented in sections 1 and 2. The systematic procedure synthesis strategy is then outlined in section 3. A simple storage system is adopted in this section as an example for illustration convenience. To further demonstrate the feasibility and correctness of the proposed approach, additional examples of the realistic air-drying process [34,35] and more complicated three-tank storage system [36,37] are reported in detail in section 4. Finally,

†To whom correspondence should be addressed.
E-mail: ctchang@mail.ncku.edu.tw

conclusions are given at the end of this paper.

GENERAL FRAMEWORK OF AUTOMATON MODEL

To facilitate clear description of the proposed method, a brief summary of the automaton structure is first given here. Specifically, a deterministic automaton A can be regarded as a sixtuple:

$$A=(X, E, f, \Sigma, x_0, X_m) \quad (1)$$

where, X is the set of system states; E is the event set; f: X×E→X represents the transition function; Σ: X→2^E denotes the active event function and 2^E is the power set of E (i.e., the set of all possible subsets of E); $x_0 \in X$ is the initial system state; $X_m \subseteq X$ is the set of marked states. The transition function f(x, e)=x′ means that a transition from state $x \in X$ to another state $x' \in X$ is caused by the feasible event $e \in E$, while the active event function $\Sigma(x)$ can be regarded as the set of active events at state x. Notice that every automaton can also be viewed as a *language-generating machine*. The events in set E should be regarded as the *alphabets* of this language and an event sequence allowed in automaton is regarded as a trace, string or word (*trace* is used in this work). The event set E can be further partitioned into subsets of controllable and uncontrollable events: $E=E_c \cup E_{uc}$. The events in E_c are those that can be forbidden with a controller, whereas the events in E_{uc} are bound to occur in due course.

In the supervisory control paradigm (see Fig. 1), the uncontrolled plant is modeled as an automaton P and the supervisory controller S is viewed as a *mapping* or *function* from the language generated by P to the power set of E, i.e.,

$$S: L(P) \rightarrow 2^E \quad (2)$$

where, L(P) represents the set of all traces produced from automaton P. If $t \in L(P)$, then S(t) should be interpreted as the control actions allowed after executing t. It should be noted that the plant automaton may generate "illegal" traces because they are physically inadmissible, e.g., an attempt to fill a tank when it is full, or they violate a desired sequence of events, e.g., an attempt to heat a vessel when it is empty. To eliminate these unacceptable traces in L(P), a set of control specifications (which can be modeled respectively with automata $H_{spec,\,i}$ and i=1, 2, ⋯) should be introduced to restrict the system behavior to be within an admissible subset $L_a \subset L(P)$. An admissible supervisor can then be constructed accordingly to ensure $L(S/P) \subseteq L_a$, where L(S/P) represents the set of all traces obtained from the closed-loop system in Fig. 1.

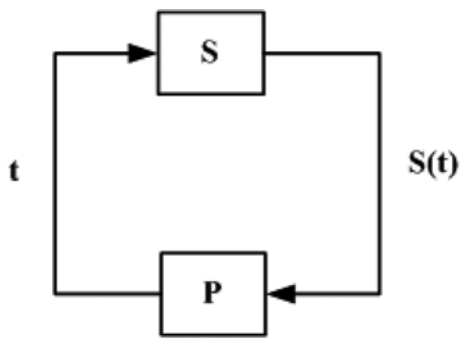

Fig. 1. Feedback loop of a supervisory control system [24,25].

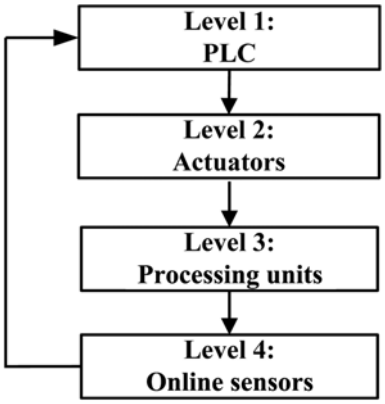

Fig. 2. Hierarchical structure of a batch process.

HIERARCHICAL STRUCTURE OF BATCH PROCESSES

As mentioned before, every batch process can be precisely described with a P&ID and a sequential function chart (SFC). All identifiable hardware items in P&ID can be treated as components of the given system and classified into a 4-level hierarchy (see Fig. 2). The top-level component is usually a programmable logic controller (PLC) used for executing the given SFC to alter the actuator states in the next level. More than one actuator may be present in the batch process, e.g., control valves, hand valves, pump, compressor, and switches, etc. These actuators are installed for the purpose of adjusting the process configuration, i.e., the material and/or energy flow patterns in the given system. Every major processing unit in P&ID, such as heat exchange, separation, reaction and storage, is considered as a level-3 component, while every on-line sensor is treated as a component in level 4. Finally, the above hierarchical structure can be viewed as a modified version of that given in ISA-S88 [38], and the current framework is adopted mainly for the convenience in model building.

The P&ID of an *uncontrolled* batch process is assumed to be given in the synthesis problems studied in this work, while the SFC is not available. Our research goal is to systematically generate proper SFC(s) to satisfy some prescribed control specifications.

SYSTEMATIC SYNTHESIS PROCEDURE

The batch operating procedures can be produced in four distinct steps:

(1) Build the automaton models of all components in the uncontrolled plant;

(2) Construct automata to represent the control specifications;

(3) Combine all automata created in steps (1) and (2) to create an *admissible* supervisor;

(4) Produce an *implementable* supervisor by augmenting the admissible supervisor with auxiliary automata and then identify the most efficient operating procedure accordingly.

A flowchart is also presented in Fig. 3 to provide an unambiguous summary of the proposed procedure synthesis strategy.

1. Component Models

Let us use the simple example given in Fig. 4 (which will be re-

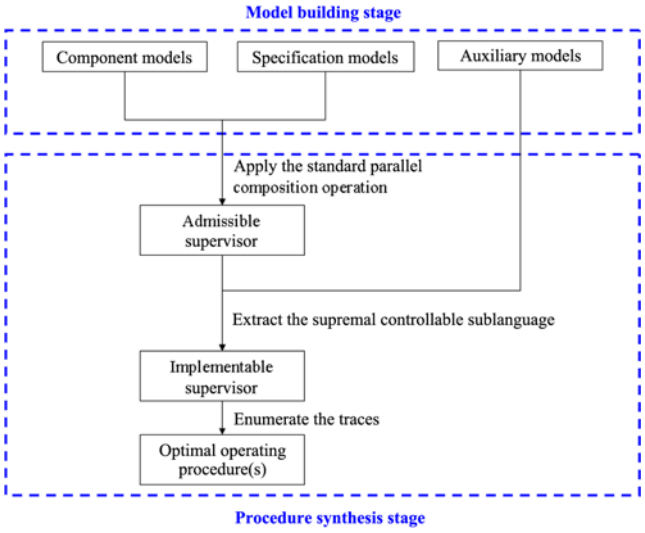

Fig. 3. Procedure synthesis strategy.

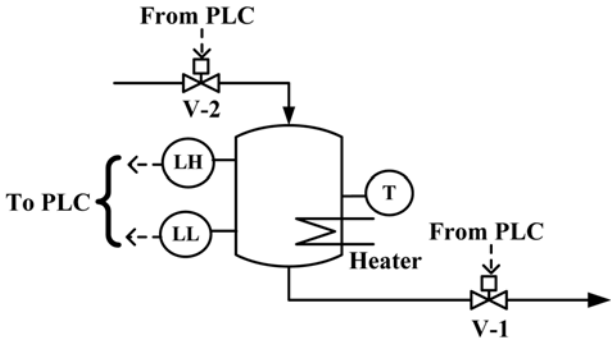

Fig. 4. A storage tank equipped with heating equipment (example 1).

ferred to as Example 1 in this paper) to illustrate the aforementioned procedure. The height of liquid level in tank is monitored with a level sensor and two distinct sensor signals, i.e., (1) *LH* (level high) and (2) *LL* (level low), may be issued to trigger the material transfer operations. In addition, the liquid temperature is also measured and two corresponding states, *TH* (temperature high) and *TL* (temperature low), may be reported to respectively activate and deactivate the heating operation. The inlet valve V-2 should obviously be open in tank-filling operation under the condition that the outlet valve V-1 is kept closed. On the other hand, V-1 should be open while keeping V-2 closed if tank draining is required. To facilitate error-free operations, liquid should be transported into or from the tank only when the corresponding sensor signal, *LL* or *LH*, can be confirmed. Notice also that the heater should be turned off after reaching the target temperature and it should not be turned on before the tank is *LH*. Initially, the liquid level and the temperature in tank are at *LL* and *TL* respectively, both valves are closed, and the heater is off. The automaton models of *uncontrolled* plant components are outlined in the sequel:

- Level 2: It should be first noted that the process configuration of a batch system is governed by the collective states of actuators. Although there are three actuators in Fig. 4 and, therefore, eight possible configurations, only four of them can be allowed and they are

Table 1. Actuator combinations in example 1

V-1 state*	V-2 state*	Heater state*	Symbol
C	C	C	GV_{01}
C	C	O	GV_{02}
C	O	C	GV_{03}
O	C	C	GV_{04}

*O and C denote open/on and close/off positions respectively

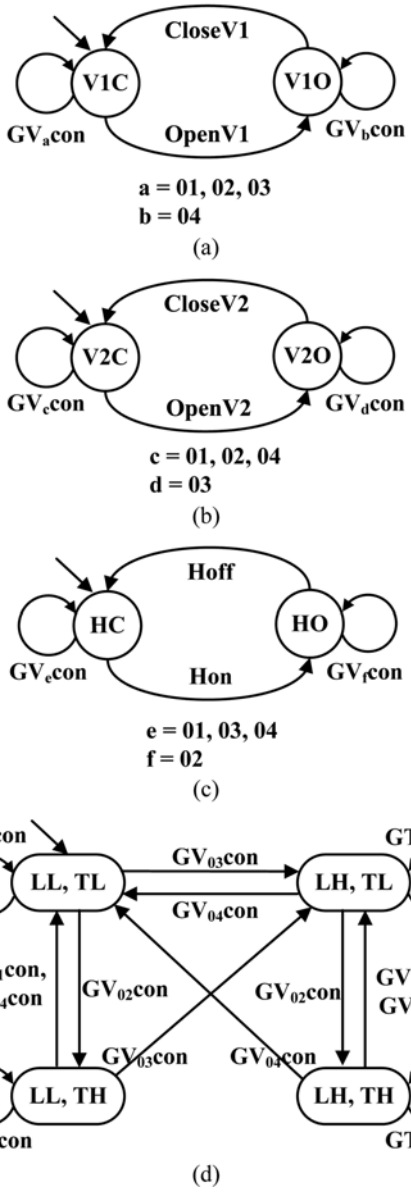

Fig. 5. Component models in example 1: (a) valve V-1 model; (b) valve V-2 model; (c) heater model; (d) tank model associated with level and temperature.

referred to in this paper as GV_{01}, GV_{02}, GV_{03} and GV_{04}, respectively (see Table 1). The actuator models can be built on the basis of these allowed configurations. The automaton representation of V-1 can be found in Fig. 5(a). States *V1C* and *V1O* in this model are used to represent the close and open positions, respectively, while the

events *openV*1 and *closeV*1 denote the corresponding close-to-open and open-to-close processes. In addition, the transitions $GV_a con$ and $GV_b con$, respectively, represent V-1 continues at close and open position for a sufficiently long period of time. Notice also that a similar model for V-2, which is shown in Fig. 5(b), can be established with essentially the same approach. The heater model is presented in Fig. 5(c). States *HF* and *HN* here are used to represent the off and on positions, respectively, while the events *Hon* and *Hoff* denote the corresponding close-to-open and open-to-close processes. Finally, the events $GV_e con$ and $GV_f con$ represent the heater continues at off and on positions, respectively.

- Level 3: The liquid transfer and heating operations in tank can be modeled with the automaton in Fig. 5(d). It can be observed that four tank states, (*LL*, *TL*), (*LL*, *TH*), (*LH*, *TL*) and (*LH*, *TH*), are considered here, and $GT_{01} con$, $GT_{02} con$, $GT_{03} con$ and $GT_{04} con$ are the corresponding events denoting the tank continues at these four states, respectively. Notice that all process configurations defined previously should result in state transitions. For instance, $GV_{03} con$ should facilitate the (*LL*, *TL*)-to-(*LH*, *TL*) process and $GV_{04} con$ should cause the (*LH*, *TL*)-to-(*LL*, *TL*) process.
- Level 4: For the sake of brevity, the sensor model is omitted in the present study and the online measurements are considered to be identical to the tank states.

2. Specification Representations

In this present study, the control specifications are used to ensure safety and/or operability. Specifically, it is used to achieve or forbid a prescribed event/state sequence to avoid physically inadmissible behaviors, e.g., filling a tank when it is full, heating a vessel when it is empty, etc. Various automata can be constructed to satisfy these requirements [33].

The control specifications for Example 1 can be summarized as follows:

- Spec 1: Discharge the tank and switch on the heater while liquid level is at *LH* and fill it while level is at *LL* (see Fig. 6(a)). Notice that $GT_g con$ and $GT_h con$ denote the events that liquid level is maintained at *LL* and *LH*, respectively.
- Spec 2: Switch off the heater when liquid temperature is at *TH* while switch on the heater and close V-2 when temperature is at *TL* (see Fig. 6(b)). Notice that $GT_i con$ and $GT_j con$ denote the events that liquid temperature is maintained at *TH* and *TL*, respectively.
- Spec 3: Avoid opening both valves simultaneously (see Fig. 6(c)).
- Spec 4: Avoid heating before V-1 is closed (see Fig. 6(d)).
- Spec 5: Avoid opening V-1 before heater is switched off (see Fig. 6(e)).
- Spec 7: Specify event sequence, i.e., E_{01} E_{02} E_{03}. Notice that event E_{01} denote actuator actions, whereas E_{02} and E_{03} represent process configurations and combined states of processing units, respectively (see Fig. 6(f)).

3. Supervisor Generation

An admissible supervisor shown in Fig. 7 can be produced by applying the standard *parallel composition* operation on the aforementioned component models and specifications models. The practically implementable supervisor can then be identified by extracting the *supremal controllable sublanguage* from this admissible supervisor. For this purpose, two auxiliary automata can be constructed to define the target state(s) or event(s) and also to set the upper limit on the total number of actuator actions. Let us again consider Example 1 here for illustration convenience:

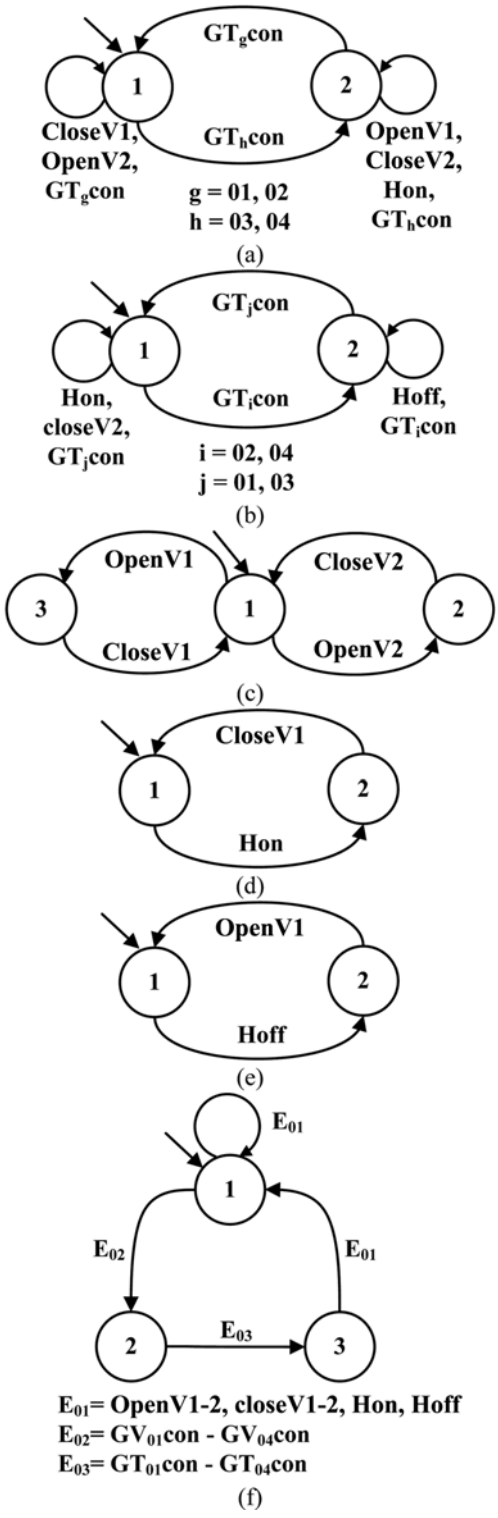

Fig. 6. Specification models in example 1: (a) Spec 1; (b) Spec 2; (c) Spec 3; (d) Spec 4; (e) Spec 5; (f) Spec 6.

- The auxiliary automaton in Fig. 8(a) is adopted for the purpose of specifying termination mechanism, i.e., stopping operation after events $GV_{03} con$ and $GT_{03} con$ twice. Thus, state should be marked as the operation target and disallow any events to occur later.

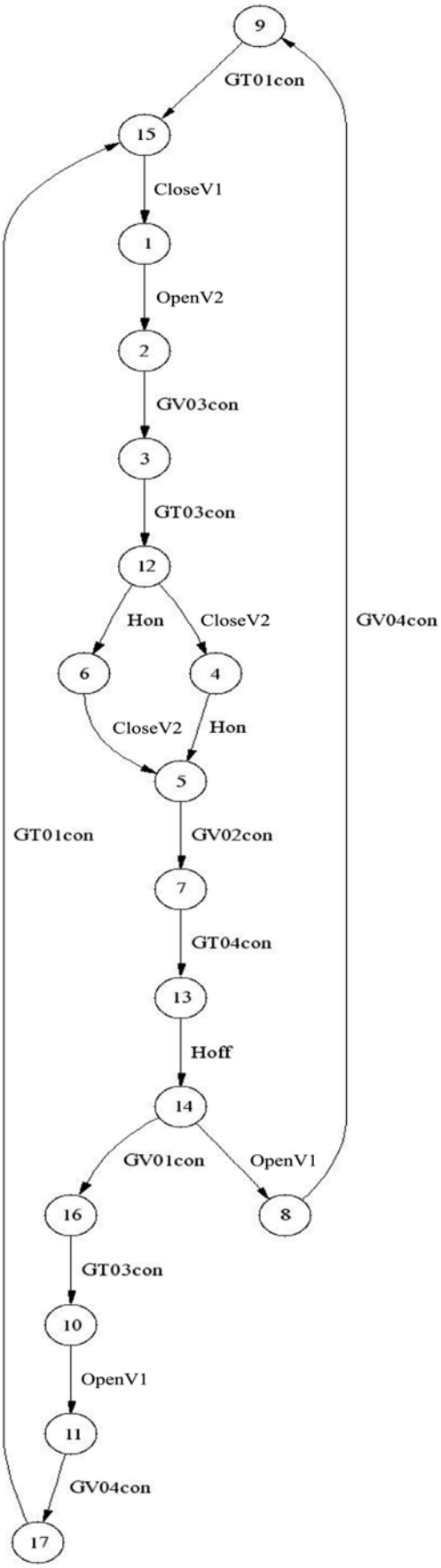

Fig. 7. Admissible supervisor for example 1.

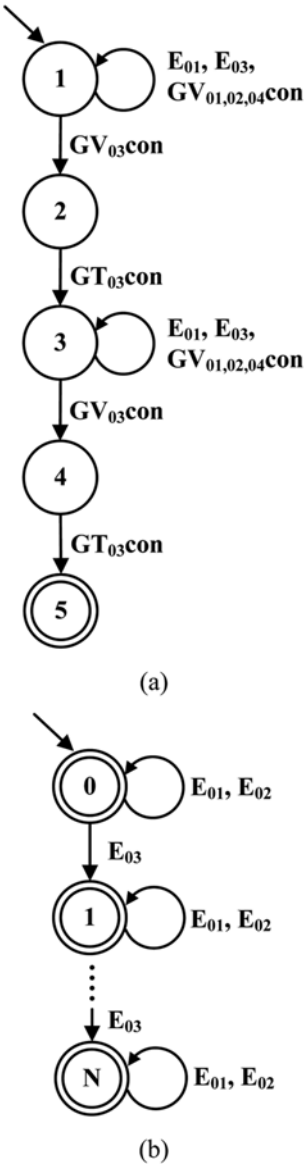

Fig. 8. Auxiliary automata in example 1: (a) Terminating the admissible supervisor after $GV_{03}con$ and $GT_{03}con$ twice; (b) Limiting the number of operation steps.

• The auxiliary automaton in Fig. 8(b) is used to limit the total number of operation steps. Since the initial state is driven to state n (n=0, 1, ⋯, N) after n operation steps, the maximum number of operation steps, N, in the operation procedure(s) can be imposed by augmenting the admissible supervisor with this automaton. Notice also that, in order to allow fewer operation steps to be taken in the operation procedure(s), all states are marked in this model. Consequently, this auxiliary automaton facilitates easy identification of all feasible procedures with n≤N operation steps and also the most efficient one(s) among them.

The above two models have been incorporated into the admissible supervisor in Fig. 7 to generate the implementable supervisor in Fig. 9.

4. Procedure Synthesis

Having created the implementable supervisor, the correspond-

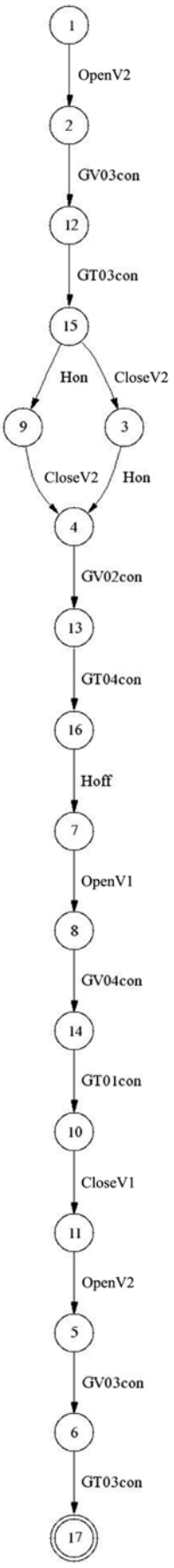

Fig. 9. Implementable supervisor for example 1.

ing operating procedures can then be identified accordingly. This task should be performed in the following stpes:

1. Classify the supervisor events into three sets, E_{AC}, E_{PC} and E_{AA}, which are associated with the *activation conditions*, *process configurations* and *actuator actions*, respectively;

2. Produce a *reduced* supervisor by grouping the consecutive actuator actions (the events in E_{AA}) between activation conditions (the events in E_{AC}) and *process configurations* (the events in E_{PC}). It is assumed that the grouped events occur simultaneously;

3. Remove all events in E_{PC}, which are treated as unobservable events.

4. Identify all possible procedures by enumerating traces generated from the reduced supervisor.

Let us consider the implementable supervisor shown in Fig. 9 as an example. Notice that actuator actions *openV*2 and *Hon* between events E_{AC} and E_{PC} should be grouped together and treated as simultaneous events. The reduced implementable supervisor can then be obtained for Example 1. The most efficient operating procedure, the ones with fewest operation steps, can be enumerated accord-

Table 2. Identified SFC in example 1: (a) Operation steps; (b) Activation conditions

(a)

Operation step	Control actions
S_0	Initialization
S_1	(1) Close V-1
	(2) Open V-2
S_2	(1) Switch on Heater
	(2) Close V-2
S_3	(1) Switch off Heater
	(2) Open V-1

(b)

Symbol	Conditions
T_1	Start
T_2	LH
T_3	TH
T_4	TL & LL

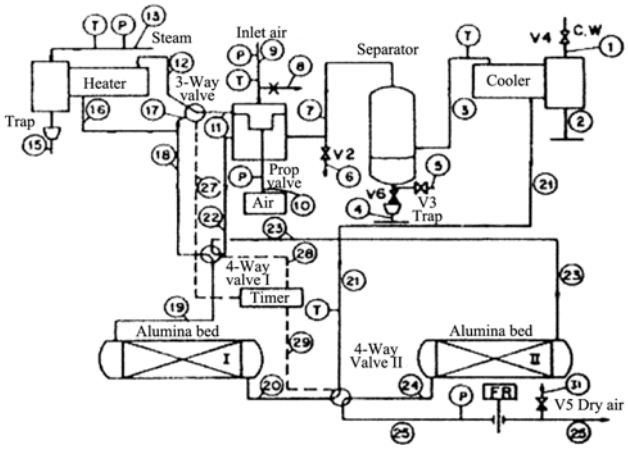

Fig. 10. A utility air drying process [34,35] (example 2).

ingly. The operation steps and activation conditions of the corresponding SFC are listed in Table 2(a) and Table 2(b), respectively.

APPLICATIONS

To demonstrate the feasibility and effectiveness of the proposed approach, two additional case studies have been carried out in this work. The first is concerned with an air-drying process, while the other a three-tank storage system.

1. Example 2

An air-drying process [34,35] is considered here to test the proposed method. The process flow diagram of this system can be found in Fig. 10. For the sake of completeness, a brief process description is provided in the sequel.

Ambient air, which contains water vapor, enters the process in stream 9 and the air passes through a bed of alumina, where the water vapor is adsorbed. The dried air leaves in stream 25. Two beds are used to maintain a continuous supply of dry air. When one bed is removing water from air, the other is being regenerated and then cooled. Since a saturated bed cannot be employed for dehumidification purpose, the regeneration operation should be executed to introduce hot air in the saturated beds to strip water from the alumina. The regenerated bed must then be cooled with the inlet air before returning to the air-drying operation. Both beds experience the same operation cycle. It is assumed that the in-service adsorption bed reaches the full saturation level during the two periods when the stripping and cooling operations are performed on the other bed. Thus, the states of each alumina bed can be characterized with two distinct parameters: the bed temperature and water content. It is assumed that both parameters can be measured online. Two distinct temperature states--high and low--and three separate water-content levels--unsaturated, half-saturated and saturated--are considered in this example. The initial bed temperature and water content of B-1 are assumed to be low and saturated, respectively, while those of B-2 are low and unsaturated, respectively.

The states of three valves, the 3-way valve 3-V and the two 4-way valves 4-V-I and 4-V-II, determine the system configuration. Every valve in this system can only be switched to two alternative positions:. "+" and "−". The relationships between the valve positions and the stream flows are shown in Table 3. The position of 3-V governs the route of inlet air flow, namely, the fresh air can either be directed to the heater or simply bypass it. The position of 4-V-I defines the connections between the alumina beds and their air supplies. The air consumed in each bed can be taken either from the lower port of proportioning valve (for dehumidification) or from

Table 3. Relationships between valve positions and stream flows in Example 2 [35]

Valve	Valve position	Stream flow
3-V	+	11->12
	−	11->17
4-V-I	+	18->19 and 22->23
	−	18->23 and 22->19
4-V-II	+	20->21 and 24->25
	−	20->25 and 24->21

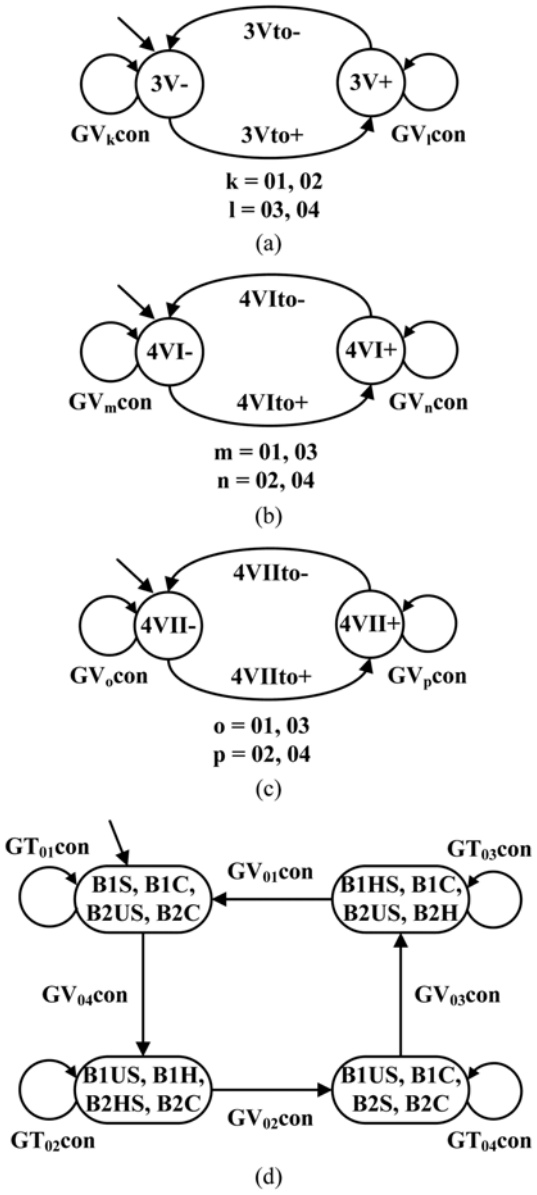

Fig. 11. Component models in Example 2: (a) 3-way valve model; (b) 4-way valve I model; (c) 4-way valve II model; (d) bed I and bed II model associated with water content and temperature.

system inlet (for regeneration or cooling). The position of valve 4-V-II determines the destinations of the exit airs from these two beds: the air can be either discharged or recycled. Initially, all three valves are assumed to be at the "−" position.

According to the proposed procedure, the automaton representations of plant components can be established and they are presented in Figs. 11(a)-(d). Although there are three actuators in Fig. 10 and eight possible configurations in this system, only four are allowed and they are referred to as GV_{01}, GV_{02}, GV_{03} and GV_{04}, respectively (see Table 4). The automaton model of 3-V is given in Fig. 11(a). States 3V− and 3V+ in this model are used to represent the − and + positions, respectively, while the events 3Vto− and 3Vto+ denote the corresponding +-to-− and −-to-+ processes. In addition, the transitions $GV_k con$ and $GV_l con$, respectively, represent 3-V con-

tinues at – and + position for a sufficiently long period of time. Notice also that a similar model for 4-V-I and 4-V-II, which are shown in Fig. 11(b) and Fig. 11(c), can be developed with essentially the same approach. The bed temperatures and water contents in B-1 and B-2 can both be described with the automaton in Figs. 11(d). It can be observed that only four different bed states, (*B1S*, *B1C*, *B2US*, *B2C*), (*B1US*, *B1H*, *B2HS*, *B2C*), (*B1US*, *B1C*, *B2S*, *B2C*) and (*B1HS*, *B1C*, *B2US*, *B2H*), are considered here. Notice that *B1US*, *B1HS* and *B1S* represent three increasing water-content levels in B---unsaturated, half-saturated and saturated--while *B1C* and *B1H* denote low and high bed temperatures, respectively. Events $GT_{01}con$, $GT_{02}con$, $GT_{03}con$ and $GT_{04}con$ in this model are used to represent the situations when both beds are maintained at the corresponding states.

Table 4. Combinations of valve positions in example 2

V-3	V-4-I	V-4-II	Symbol
–	–	–	GV_{01}
–	+	+	GV_{02}
+	–	–	GV_{03}
+	+	+	GV_{04}

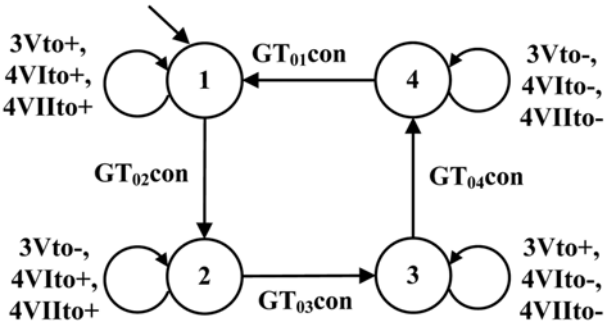

Fig. 12. Specification model (example 2).

Table 5. Identified SFC in example 2: (a) Operation steps; (b) Activation conditions

(a)

Operation step	Control actions
S_0	Initialization
S_1	(1) Switch 3-V to +; (2) Switch 4-V-I to +; (3) Switch 4-V-II to +
S_2	Switch 3-V to –
S_3	(1) Switch 3-V to +; (2) Switch 4-V-I to –; (3) Switch 4-V-II to –
S_4	Switch 3-V to –

(b)

Symbol	Conditions
T_1	Start
T_2	B1US & B1H & B2HS
T_3	B1C & B2S
T_4	B1HS & B2US & B2H
T_5	B1S & B2C

It should be pointed out that every allowed process configuration results in a state transition process. For example, $GV_{04}con$ could

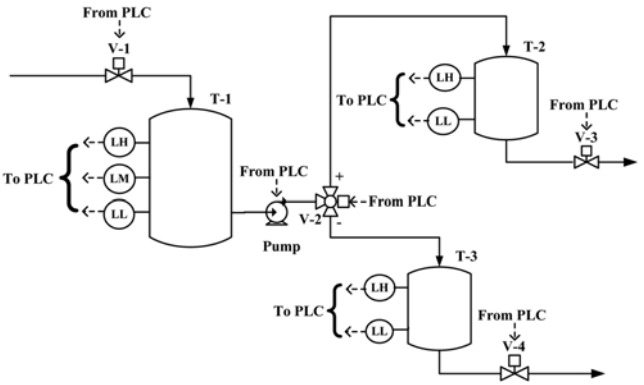

Fig. 13. A three-tank storage system [36,37] (example 3).

Table 6. Allowed process configurations in example 3

V-1	V-2	V-3	V-4	Pump	Symbol
C	–	C	C	O	GV_{01}
C	–	C	O	C	GV_{02}
C	–	O	C	C	GV_{03}
C	–	O	C	O	GV_{04}
C	–	O	O	C	GV_{05}
C	+	C	C	O	GV_{06}
C	+	C	O	C	GV_{07}
C	+	C	O	O	GV_{08}
C	+	O	C	C	GV_{09}
C	+	O	O	C	GV_{10}
O	–	C	C	C	GV_{11}
O	–	C	O	C	GV_{12}
O	–	O	C	C	GV_{13}
O	–	O	O	C	GV_{14}
O	+	C	C	C	GV_{15}
O	+	C	O	C	GV_{16}
O	+	O	C	C	GV_{17}
O	+	O	O	C	GV_{18}

Table 7. Possible combinations of tank states in example 3

T-1	T-2	T-3	Symbol
L	L	L	GT_{01}
L	L	H	GT_{02}
L	H	L	GT_{03}
L	H	H	GT_{04}
M	L	L	GT_{05}
M	L	H	GT_{06}
M	H	L	GT_{07}
M	H	H	GT_{08}
H	L	L	GT_{09}
H	L	H	GT_{10}
H	H	L	GT_{11}
H	H	H	GT_{12}

facilitate the transition from state (*B1S*, *B1C*, *B2US*, *B2C*) to state (*B1US*, *B1H*, *B2HS*, *B2C*). Finally, notice that a similar representation can also be adopted to model B-2.

The control specification can be summarized in a general statement such that any actuator action can be taken only after the occurrence of a specific level-3 event. As an example, 3-V, 4-V-I and 4-V-II can be switched to the "–", "+", "+" positions only after $GT_{02}con$. The corresponding specification model is given in Fig. 12.

By applying the standard parallel composition operation [33] on the aforementioned component models and specifications models, an admissible supervisor can be produced and a suitable operating procedure can be identified accordingly.

The operation steps and activation conditions of the corresponding SFC are listed in Table 5(a) and Table 5(b), respectively. This procedure is actually the same as that reported in Wang et al. [35]. Also, only five alternative sets of sensors are allowed for use as activation conditions in this procedure: (1) the bed temperature and water content of B-1; (2) the bed temperature and water content of B-2; (3) the water contents of both B-1 and B-2; (4) the bed temperatures of both B-1 and B-2; (5) the bed temperature and water

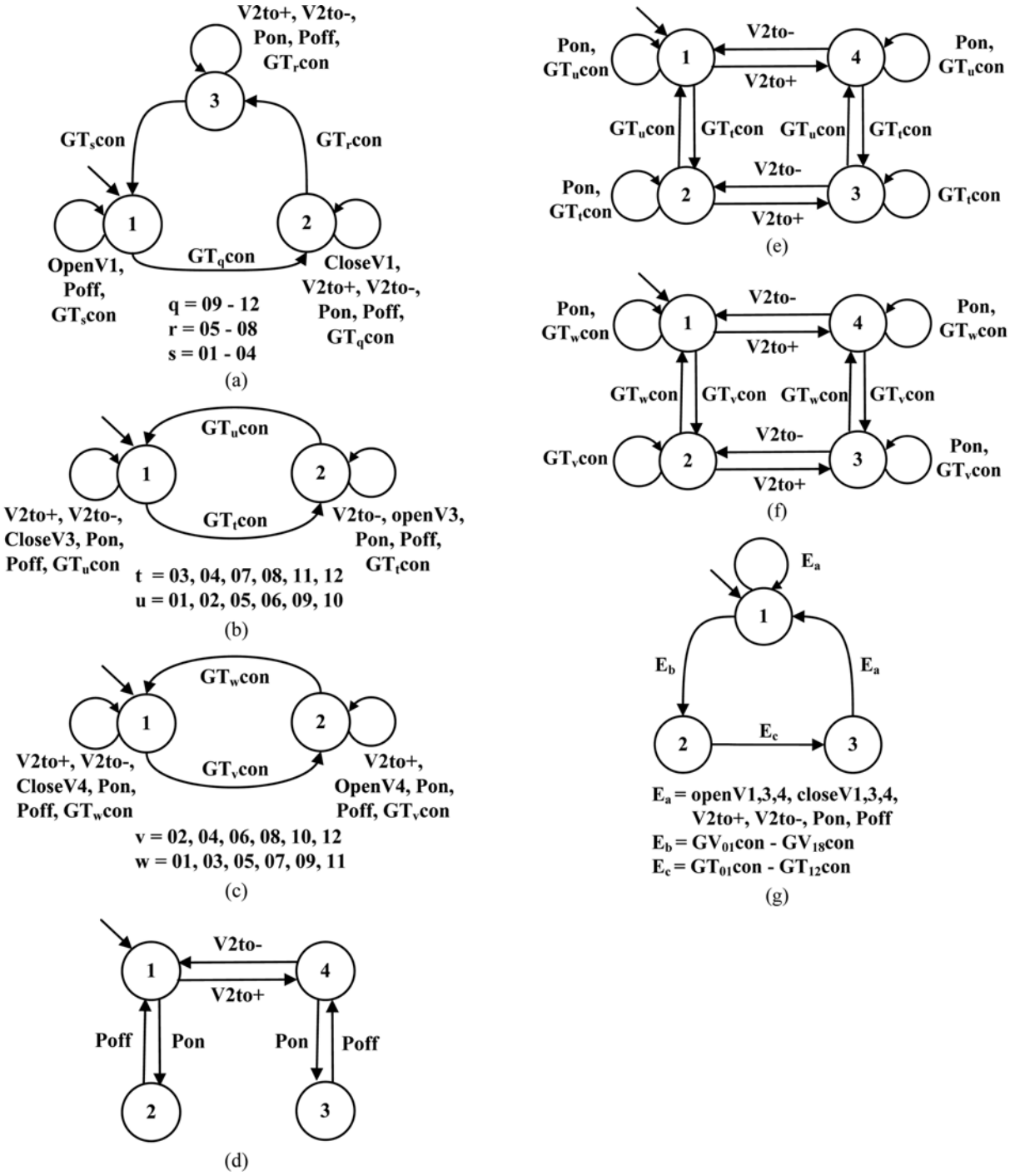

Fig. 14. Specification models in Example 3: (a) Spec 1; (b) Spec 2; (c) Spec 3; (d) Spec 4; (e) Spec 5; (f) Spec 6; (g) Spec 7.

content of both B-1 and B-2. Note that there are two sensors in sets (1)-(4), while four are needed in set (5).

2. Example 3

Let us next consider the three-tank storage system presented in Fig. 13 [36,37]. Valve V-1 in this system is used to fill tank T-1, whereas valves V-3 and V-4 are used to discharge the material in tanks T-2 and T-3, respectively. Notice that a pump is installed on the outlet pipeline of T-1, which is connected to a 3-way valve V-2. The material in T-1 can be transferred into tanks T-2 and T-3 by switching V-2 to the "+" and "–" positions, respectively. Tank T-1 is equipped with a level sensor and this sensor reports three conditions: (1) *LL* (level low); (2) *LM* (level middle); (3) *LH* (level high). The level sensors on tanks T-2 and T-3 are used to detect only two conditions: (1) *LL* (level low) and (2) *LH* (level high). It is assumed that tank T-1 can be filled only after sensor signal *LL* is issued, and discharged only after *LH* is reached. On the other hand, tanks T-2 and T-3 can obviously be discharged when level is at *LH* and filled at *LL*. For safety reason, valve V-2 is allowed to be operated only when the pump is not running. The initial conditions adopted in this example can be summarized below:

(1) the pump is switched off,

(2) all tanks are empty,

(3) valve V-2 is at the position "–", and

(4) all other valves are closed.

Since there are five actuators in Fig. 13, 32 possible configurations can be enumerated. Among them, only 18 are allowed in this system (see Table 6). In a similar fashion, 12 possible combinations of the tank states can be enumerated (see Table 7) and all of them are permissible here. The automaton representations of plant components can be built accordingly by following the proposed procedure. For the sake of brevity, the implementation steps are not repeated here and the resulting models are also omitted.

The required control specifications are outlined as follows:

Spec 1: Any actuator action can be taken after the occurrence of a specific event in tank T-1, e.g., open V-1 after *T1Lcon*. Notice that $GT_q con$, $GT_r con$ and $GT_s con$ denote the events that T-1 is maintained at *LH*, *LM* and *LL*, respectively.

Spec 2: Any actuator action can be taken after the occurrence of a specific event in tank T-2, e.g., close V-3 after *T2Lcon*. Notice that $GT_t con$ and $GT_u con$ denote the events that T-2 is maintained at *LH* and *LL*, respectively.

Spec 3: Any actuator action can be taken after the occurrence of a specific event in tank T-3, e.g., close V-4 after *T3Lcon*. Notice that $GT_y con$ and $GT_w con$ denote the events that T-3 is maintained at *LH* and *LL*, respectively.

Spec 4: Avoid switching V-2 when the pump is on.

Spec 5: Avoid turning on pump after *T2Hcon* and V2to+.

Spec 6: Avoid turning on pump after *T3Hcon* and V2to–.

Spec 7: Impose event sequence $E_a\ E_b\ E_c$. Event E_a denotes an actuator action, whereas E_b and E_c represent an allowed process configuration and a possible combination of tank states, respectively.

Spec 8: Avoid opening V-1 and turning on pump simultaneously.

Spec 9: Avoid opening V-3, switching V-2 to "+" and turning

Table 8. Definitions of events in Fig. 13

Events	Description
OpenV1	Open V-1
CloseV1	Close V-1
V2to+	Switch V-2 to + position
V2to–	Switch V-2 to – position
OpenV3	Open V-3
CloseV3	Close V-3
OpenV4	Open V-4
CloseV4	Close V-4
Pon	Switch pump to on position
Poff	Switch pump to off position

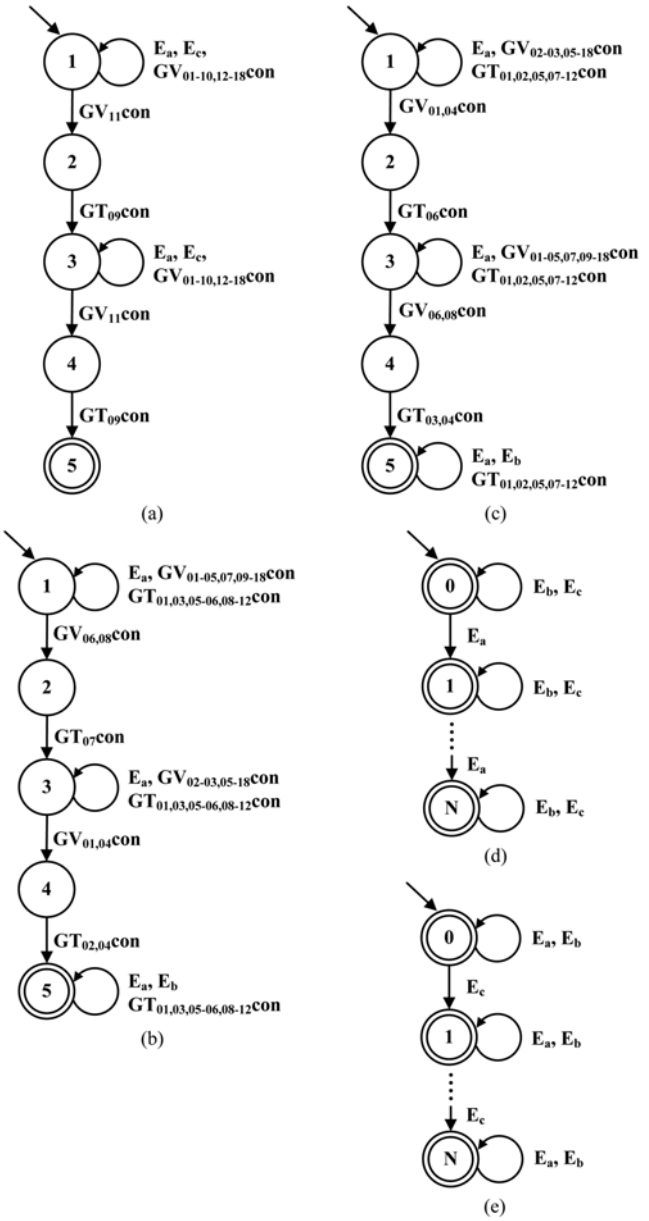

Fig. 15. Auxiliary automata in Example 3: (a) Terminating the admissible supervisor after $GV_{11}con\ GT_{09}con$ twice; (b) Specify the order of material-transfer tasks i.e., fill T-2 first and then T-3; (c) Specify the order of material-transfer tasks, i.e., fill T-3 first and then T-2; (d) Limiting the number of actuator actions; (e) Limiting the number of operation steps.

on pump simultaneously.

Spec 10: Avoid opening V-4, switching V-2 to "–" and turning on pump simultaneously.

Notice that specs 1-7 can be characterized with the automata shown in Figs. 14(a)-(g), respectively, while specs 8-10 can be built with the same approach in Fig. 6(c). For the sake of brevity, the models for specs 8-10 are omitted. Note also that the definitions of events in these models can be found in Table 8.

After assembling all component models and specification models with the parallel composition operation, an admissible supervisor can be generated. To identify the most efficient operating procedure(s), this supervisor should then be augmented with the auxiliary automata presented in Figs. 15(a)-(e). The automaton in Fig. 15(a) is adopted for the purpose of specifying a termination mechanism, namely, the operation should be stopped after triggering events $GV_{11}con$ $GT_{09}con$ twice. The automata in Fig. 15(b) and Fig. 15(c) are used to specify the two alternative orders of material-transfer tasks in the cyclic operation, i.e., material in T-1 should be transferred to fill T-2 first and then T-3 or vice versa. On the other hand, the automata in Fig. 15(d) and Fig. 15(e) are used to limit the total numbers of actuator actions and operation steps (in SFC), respectively.

Table 9. Identified SFC in Example 3: (a) Operation steps; (b) Activation conditions

(a)

Operation step	Control actions								
	SFC 1	SFC 2	SFC 3	SFC 4	SFC 5	SFC 6	SFC 7	SFC 8	SFC 9
S_0	Initialization	Initialization	Initialization	Initialization	Initialization	Initialization	Initialization	Initialization	Initialization
S_1	(1) Close V-4 (2) Open V-1	(1) Close V-3 (2) Switch V2 to – (3) Open V-1	(1) Close V-3 (2) Open V-1	(1) Close V-3 (2) Close V-4 (3) Switch V2 to – (4) Open V-1	(1) Close V-3 (2) Close V-4 (3) Open V-1	(1) Close V-3 (2) Close V-4 (3) Switch V2 to – (4) Open V-1	(1) Close V-3 (2) Close V-4 (3) Open V-1	(1) Close V-4 (2) Open V-1	(1) Close V-3 (2) Close V-4 (3) Open V-1
S_2	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch on pump	(1) Close V-1 (2) Switch V2 to + (3) Switch on pump	(1) Close V-1 (2) Switch V2 to + (3) Switch on pump	(1) Close V-1 (2) Switch V2 to + (3) Switch on pump
S_3	(1) Switch off pump (2) Open V-4	(1) Switch off pump (2) Switch V2 to + (3) Switch on pump (4) Open V-4	(1) Switch off pump (2) Switch V2 to + (3) Switch on pump (4) Open V-4	(1) Switch off pump (2) Switch V2 to + (3) Switch on pump (4) Open V-4	(1) Switch off pump (2) Switch V2 to + (3) Switch on pump (4) Open V-4	(1) Switch off pump (2) Switch V2 to + (3) Switch on pump	(1) Switch off pump (2) Switch V2 to – (3) Switch on pump	(1) Switch off pump (2) Switch V2 to – (3) Switch on pump (4) Open V-3	(1) Switch off pump (2) Switch V2 to – (3) Switch on pump (4) Open V-3
S_4	(1) Close V-4 (2) Switch on pump	(1) Switch off pump (2) Close V-4 (3) Open V-3	(1) Switch off pump (2) Close V-4 (3) Switch V2 to - (4) Open V-3	(1) Switch off pump (2) Open V-3	(1) Switch off pump (2) Switch V2 to – (3) Open V-3	(1) Switch off pump (2) Open V-3 (3) Open V-4	(1) Switch off pump (2) Open V-3 (3) Open V-4	(1) Switch off pump (2) Close V-3 (3) Open V-4	(1) Switch off pump (2) Open V-4
S_5	(1) Switch off pump (2) Open V-4								

(b)

Symbol	Conditions				
	SFC 1	SFCs 2-5	SFC 6	SFC 7	SFCs 8-9
T_1	Start	Start	Start	Start	Start
T_2	T1H	T1H	T1H	T1H	T1H
T_3	T1M & T3H	T1M & T3H	T1M & T3H	T1M & T2H	T1M & T2H
T_4	T3L	T1L & T2H & T3L	T1L & T2H	T1L & T3H	T1L & T2L & T3H
T_5	T1L & T3H	T2L	T2L & T3L	T2L & T3L	T3L
T_6	T3L				

The operation steps and activation conditions of the best SFCs, i.e., SFCs 1-9, are listed in Table 9(a) and Table 9(b), respectively. Notice that SFC 7 is the same as the procedure presented in Chen et al. [36] and also in Yeh and Chang [37]. It can be also observed that,

(1) Without stipulating the order of material transfers, SFCs 1-9 can be considered to be superior to the other alternatives. There are only ten activation conditions and five operation steps in SFC 1, while twelve actuator actions and four operation steps are needed in the remaining procedures. Notice that, T-3 is filled twice in a cyclic operation, while T-2 is never filled in SFC 1.

(2) When T-2 is required to be filled first, SFCs 7-9 should be chosen. There are twelve actuator actions and four operation steps in every alternative solution.

(3) When T-3 is required to be filled first, SFCs 2-6 become suitable candidates. There are also twelve actuator actions conditions and four operation steps in every SFC.

CONCLUSIONS

A systematic automata-based procedure is presented in this paper to synthesize all possible untimed operating procedures for any given batch chemical process. The specific steps to be performed in this procedure include: building the automaton models of the uncontrolled plant and the control specifications, constructing the admissible and implementable supervisors, and identifying the most efficient SFCs. The feasibility and correctness of this proposed approach are successfully demonstrated in three case studies in this paper. Since only manual verification procedure was adopted in these studies, future effort should therefore be devoted to the development of simulation tools to carry out more rigorous tests.

ACKNOWLEDGEMENT

This work is supported by the National Science Council of Taiwan under Grant NSC 100-2221-E-006-139-MY2.

REFERENCES

1. J. R. Rivas and D. F. Rudd, *AIChE J.*, **20**, 320 (1974).
2. I. Moon, G J. Powers, J. R. Burch and E. M. Clarke, *AIChE J.*, **38**, 67 (1992).
3. A. Sanchez and S. Macchietto, *Comput. Chem. Eng.*, **19**, S381 (1995).
4. Y. Naka, M. L. Lu and H. Takiyama, *Comput. Chem. Eng.*, **21**, 997 (1997).
5. C. Panjapornpon, M. Soroush and W. D. Seider, *Ind. Eng. Chem. Res.*, **45**, 2758 (2006).
6. J. Kim and I. Moon, *J. Loss Prevent. Proc.*, **22**, 975 (2009).
7. M. K. A. Hamid, G. Sin and R. Gani, *Comput. Chem. Eng.*, **34**, 683 (2010).
8. H. Yang, N. Li and S. Y. Li, *Asian J. Control*, **13**, 345 (2011).
9. M. L. Fravolini and G. Campa, *IEEE T. Neural Networ.*, **22**, 627 (2011).
10. C. A. Crooks and S. A. Macchietto, *Chem. Eng. Commun.*, **114**, 117 (1992).
11. S. Galán and P. I. Barton, the Annual Meeting of the American Institute of Chemical Engineers (1997).
12. H. S. Li, M. L. Lu and Y. Naka, *Comput. Chem. Eng.*, **21**, s899 (1997).
13. J. Kim, J. Kim and I. Moon, *J. Loss Prevent. Proc.*, **22**, 367 (2009).
14. R. H. Fusillo and G J. Powers, *Comput. Chem. Eng.*, **11**, 369 (1987).
15. R. Lakshmanan and G. Stephanopoulos, *Comput. Chem. Eng.*, **12**, 985 (1988).
16. S. Viswanathan, C. Johnsson, R. Srinivasan, V. Venkatasubramanian and K. E. Arzen, *Comput. Chem. Eng.*, **22**, 1673 (1998).
17. V. A. Ivanov, V. V. Kafarov, V. L. Perov and A. A. Reznichenko, *Eng. Cybern.*, **18**, 104 (1980).
18. A. Kinoshita, T. Umeda and E. O'Shima, *Proceedings of the International Symposium on Process Systems Engineering*, 114 (1982).
19. K. Hoshi, K. Nagasawa, Y. Yamashita and M. Suzuki, *J. Chem. Eng. Jpn.*, **35**, 377 (2002).
20. E. C. Yamalidou and J. C. Kantor, *Comput. Chem. Eng.*, **15**, 503 (1991).
21. S. Hashizume, T. Yajima, T. Ito and K. Onogi. *J. Chin. Inst. Chem. Eng.*, **35**, 363 (2004).
22. H. H. Chou and C. T. Chang, *Ind. Eng. Chem. Res.*, **44**, 114 (2005).
23. Y. F. Wang, H. H. Chou and C. T. Chang, *Comput. Chem. Eng.*, **29**, 1822 (2005).
24. J. W. Lai, C. T. Chang and S. H. Hwang, *Ind. Eng. Chem. Res.*, **46**, 2797 (2007).
25. P. J. Ramadge and W. M. Wonham, *SIAM J. Control Optim.*, **25**, 206 (1987).
26. P. J. Ramadge and W. M. Wonham, *P. IEEE*, **77**, 81 (1989).
27. B. A. Brandin and W. M. Wonham, *IEEE T. Automat. Contr.*, **39**, 329 (1994).
28. W. M. Wonham, *Proceedings of the IEEE international conference on industrial technology*, Goa, India, 474 (2000).
29. P. Dietrich, R. Malik, W. M. Wonham and B. A. Brandin, Implementation considerations in supervisory control. *Synthesis and Control of Discrete Event Systems*, B. Caillaud, P. Darondeau, L. Lavagno, X. Xie, Kluwer, Eds., 185 (2002).
30. P. Malik and R. Malik, *Proceedings of the 8th international workshop on discrete event systems*, Ann Arbor, Michigan, USA (2006).
31. R. J. Leduc, P. C. Dai and R. G. Song, *IEEE T. Automat. Contr.*, **54**, 1548 (2009).
32. R. Su, J. H. van Schuppen and J. E. Rooda, *IEEE T. Automat. Contr.*, **55**, 2527 (2010).
33. C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Kluwer Academic, Boston (1999).
34. J. A. Shaeiwitz, S. A. Lapp and G J. Powers, *Industrial Engineering and Chemical Process Description Development*, **16**, 529 (1977).
35. Y. F. Wang, J. Y. Wu and C. T. Chang, *Reliability Engineering and System Safety*, **76**, 91 (2002).
36. Y. C. Chen, M. L. Yeh, C. L. Hong and C. T. Chang, *Ind. Eng. Chem. Res.*, **49**, 4249 (2010).
37. M. L. Yeh and C. T. Chang, *Chem. Eng. Res. Des.*, **89**, 2652 (2011).
38. D. W. Fleming, V. A. Pillai and J. A. Pillai, *S88 Implementation Guide*, McGraw-Hill Inc., New York (1998).