CHEMICAL ENGINEERING TRANSACTIONS
VOL. 74, 2019
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.cetjournal.it
Guest Editors: Sauro Pierucci, Jiří Jaromír Klemeš, Laura Piazza

ISBN 978-88-95608-71-6; **ISSN** 2283-9216
DOI: 10.3303/CET1974112

Synthesis and Validation of Operating Procedures Based on Untimed Automata

Tai-Yu Chen, Chuei-Tin Chang*

Department of Chemical Engineering, ational Cheng Kung University, Tainan, Taiwan 70101, ROC
ctchang@mail.ncku.edu.tw

To facilitate efficient procedure synthesis, the extended finite automata are adopted in this work to model all components in a chemical process. The intended operation is divided into several stages and each characterized with a set of unique features, e.g., stable operation, condition adjustment, material charging and/or unloading, etc. The control specifications of every stage should then be stipulated accordingly and also described with automata. The observable event traces (OETs) embedded in this system can be produced by synchronizing all aforementioned automata. The candidate operating procedures are summarized with sequential function charts (SFCs) on the basis of these OETs. The commercial package ASPEN Plus Dynamic has been used to validate these SFCs in simulation studies.

1. Introduction

Despite the fact the modern chemical plants are becoming more complex than they used to be, their operating procedures are still generated manually in most cases. Manual synthesis of operating procedure in a realistic system can be a very difficult task since it is both time-consuming and error-prone. It is thus desirable to develop a systematic approach to automatically conjecture operation steps so as to achieve specific production goal (Lu et al., 2017). The untimed extended finite automata (EFA) (Åkesson et al., 2006) are utilized in the present work for such a purpose. In particular, all components in a given system are characterized with automata according to the proposed modelling principles (Li et al., 2014). To further facilitate efficient procedure synthesis, the intended operation is divided into several distinguishable stages in which the intrinsic natures of each stage, e.g., stable operation, condition adjustment, material charging and/or discharging, etc., are identifiable. The control specifications of every stage are described accordingly with automata so as to set the target state, to create different operation paths via state splitting, to limit feasible operations to those that follow only the designated partial sequences and to avoid unsafe operations by stipulating illegal strings, etc. A system model and the corresponding observable event traces (OETs) can then be generated by synchronizing all aforementioned automata via the standard function of free software SUPREMICA (Åkesson et al., 2006). For any practical application, one or more operating procedures can be easily extracted from these traces and formally summarized with sequential function charts (SFCs). The commercial package ASPEN Plus Dynamic has been used to validate these SFCs in simulation studies. A flash start-up example is reported in this paper to demonstrate the effectiveness of the proposed approach.

2. Untimed extended finite automata

To facilitating clear description of the proposed model construction method, a brief review of the automaton structure is first given here. Specifically, a deterministic untimed automaton A can be viewed as a six-tuple:

$$A=\left(X,E,f,\Sigma,x_0,X_m\right) \tag{1}$$

where, X is the set of system states; E is the event set; $f:X\times E\to X$ represents the state transition function; $\Sigma:X\to 2^E$ 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 state; $X_m\subset X$ 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 event $e\in E$, while the active event function

Paper Received: 12 June 2018; Revised: 15 October 2018; Accepted: 3 April 2019

Please cite this article as: Chen T.-Y., Chang C.-T., 2019, Synthesis and Validation of Operating Procedures Based on Untimed Automata, Chemical Engineering Transactions, 74, 667-672 DOI:10.3303/CET1974112

$\Sigma(x)$ can be regarded as the set of active events at state x. The sketch of an example automaton can be found in Figure 1. The circles ($S0$, $S1$ and $S2$) are referred to as places and they are used to represent system states, while the directed arcs denote events and $E1$, $E2$ and $E3$ are their labels. The initial state is indicated with an input arrow and the marked state is darkened. It should be noted the EFA is an improved version of the aforementioned structure. Specifically, each event in EFA is equipped with two extra auxiliary elements, i.e., variable and guard, and these elements are further explained below:

- An integer variable (with user-specified upper and lower bounds) can be used to update the equipment state after completing an event-driven transition. An example is shown in Figure 1, in which variable a is updated to 1 ($a = 1$) via event $E1$.
- A guard is the sufficient condition of the corresponding state transition. Let us again consider Figure 1 as an example and assume the initial value of variable a is 0. Therefore, only event $E1$ is permissible at the initial state $S0$ due to its guard "$a == 0$" and, when $S1$ is reached after state transition, variable a should be updated to 1.

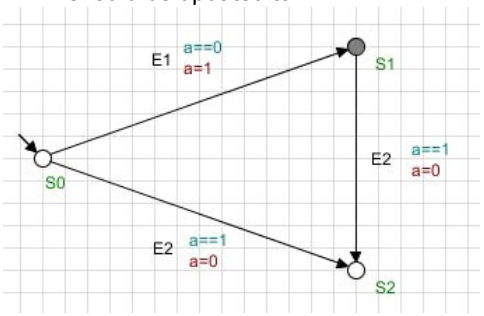

Figure 1: Graphic representation of an automaton.

3. An illustrative example

For illustration convenience, let us consider the startup operation of the continuous flash process described in the process flow diagram (PFD) in Figure 2. In this system, there are four PID controllers (FC01, TC01, PC01 and LC01) for controlling the feed rate, the temperature, the vapor pressure and the liquid level in flash drum, respectively. The corresponding actuators are control valves Vin, Vlps, Vvap and Vliq. It is assumed that, at steady state, the feed is a mixture of 30 wt% water and 70 wt% methanol and its flowrate, temperature and pressure are kept at 26000 kg/hr, 20℃ and 1.1 bar, respectively. It is also assumed that, initially, all valves are closed, all controllers are on manual and the flash drum is empty and at room temperature. Finally, it is required that the concentration of methanol in the top product should not be lower than 87 wt%.

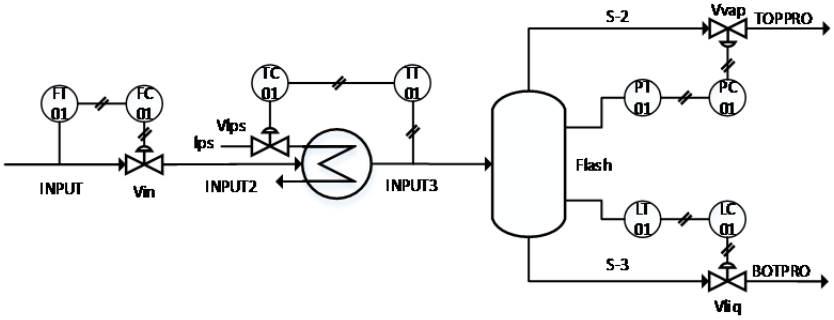

Figure 2: PFD of a continuous flash process.

4. Process structure

Basically every identifiable hardware item in the PFD is treated as a component in this study and they are classified into a 5-level hierarchy according to Figure 3 below. The top-level component, i.e., supervisor, is usually a programmable logic controller (PLC) or human operator; The PID controllers and actuators are classified as the second-level components; The material and energy flows surrounding each unit in the subsequent level are viewed as the third-level components; Every processing unit, such as the flash drum, is regarded as a fourth-level component; All online sensors (i.e., FT01, TT01, PT01 and LT01 in Figure 2) are grouped into level 5.

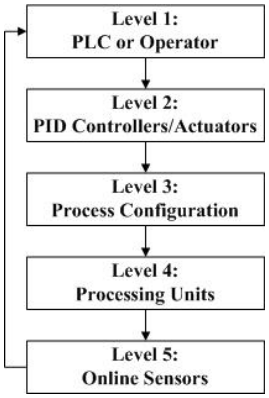

Figure 3: Hierarchical structure of a chemical process.

5. Component model

Every component in a given process is modelled with an automaton in this study. To build such a model for any component, all its normal and failed states should be first enumerated and represented with distinct places. The initial state should also be selected and the corresponding place indicated with an incoming arrow, but there are no needs to assign the marked states in the component model. All normal and failure events that facilitate state transitions should then be identified and each represented with a directed arc between the input and output places. Finally, the guard(s) of every event and the resulting variable value(s) should be added on the corresponding arc. As an example, the component model of control valve Vin is presented in Figure 4.

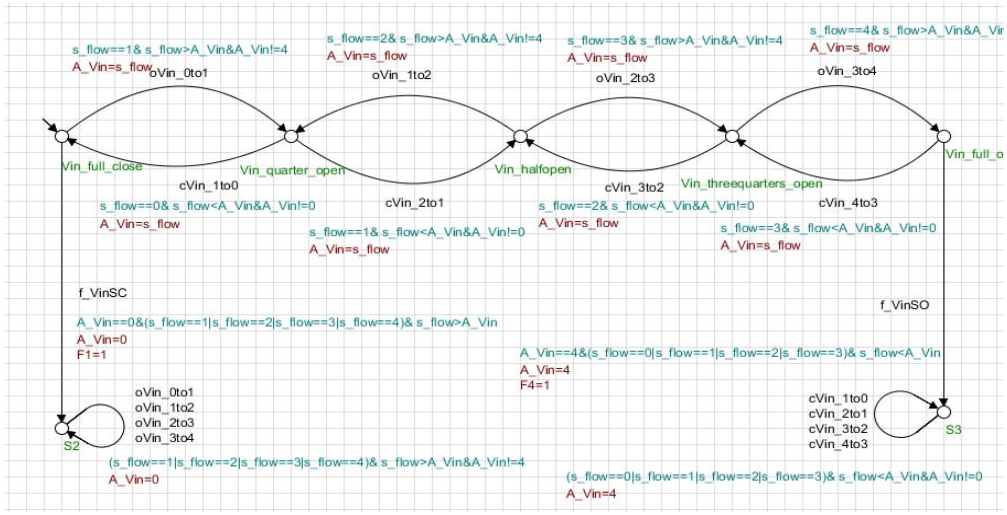

Figure 4: Component model of control valve Vin.

6. Path explosion

Intuitively, the system model for the flash startup process can be constructed by synchronizing all aforementioned automata with an automaton that specifies the final target of operation (see Figure 5). Since only the generic engineering knowledge is utilized to build the component models, an overwhelmingly large number of paths may be extracted from this integrated automaton even when the system dynamics is moderately complex.

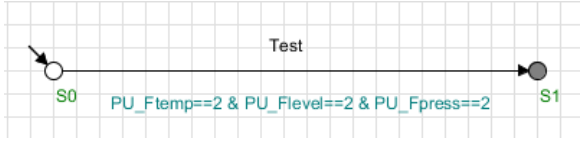

Figure 5: Final target of flash startup operation.

In particular, the aforementioned synchronization operation yielded the complicated path network in Figure 6 for the flash startup example. Since the dynamic behavior of a MIMO system cannot be adequately described with the untimed automata developed on the basis of qualitative information, this network consists of not only one or more feasible path but also an extremely large number of unnecessary and impractical scenarios. In other words, some of the paths generated by SUPREMICA may not be real and, furthermore, the operationally infeasible deadlocks and/or livelocks may even be present in this unrealistic path network.

Figure 6: Path explosion.

7. Intrinsic stages and their control specifications

Although the final goal of an operation is usually unambiguously given, this goal can only be reached via a series of stages which are not explicitly specified a priori. It is thus important to uncover these intrinsic stages and identify their unique features in advance. Generally speaking, these features may include: (1) material charging, (2) material unloading, (3) reaction, (4) state transfer, (5) phase change and (6) stable operation, etc. Clearly, each stage in an operation must be characterized with a unique set of features. For illustration purpose, let us revisit the flash startup process. In the initial stage (stage 1), it is necessary to place a small quantity of raw material in the flash drum and allow the liquid level reaching a height which is safe for heating. In the next stage (stage 2), the temperature and pressure in the drum should be raised to the set points and the input and output flow rates should be adjusted to the steady-state levels. Finally, a stable operation should be maintained automatically with the PID controllers in stage 3. Thus, the feature sets of the above three stages may be listed as follows: (1) state transfer and material charging, (2) state transfer, phase change and material charging and unloading, and (3) stable operation. All features in a stage should be described with the so-called “control specifications” and the corresponding automata. Five different types of automata may be constructed for use to set the target state (type A), to perform state splitting (type B), to impose a partial sequence (type C), to suppress an illegal substring (type D) and to ensure alternation between two particular events (type E). Due to space limitation, Figures 7 – 9 below only show the automata needed for depicting the control specifications of stage 1.

- Type A: Since the operational goal of stage 1 is to place a small amount of liquid mixture in the flash drum before heating, this target is specified in Figure 7 as the guard of event $\text{step}\,1$. Notice the prerequisite condition $\text{PU\_Flevel} == 1$ denotes that the liquid level must reach the discrete value of 1. Notice also that $S0$ and $S1$ represent the initial and marked states, respectively.

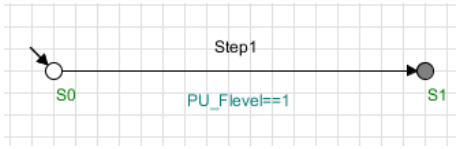

Figure 7: Control specification of type A for stage 1.

- Type C: It is also desired to minimize the charging time by maximizing the feed rate in stage 1. The corresponding control specification can be modelled by the automaton in Figure 8. Notice the guard of event Vin_full_open is $\text{A\_Vin} == 4$ (the largest opening of valve Vin is at the discrete value of 4). In this example this is the condition when Vin is fully open.

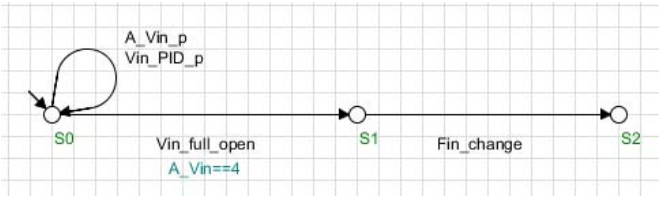

Figure 8: Control specification of type C for stage 1.

- Type D: The illegal strings specified in Figure 9 are adopted to prevent inlet valve (Vin) closing and heater (H) switching on during stage 1. Notice that, in SUPREMICA, the guard $1 == 0$ is used to forbid the self-looping events, i.e., $\mathrm{Vin\_PID\_n}$ and $\mathrm{H\_PID\_p}$, on $\mathrm{S1}$.

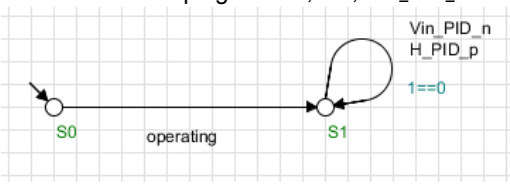

Figure 9: Control specification of type D for stage 1.

8. Procedure synthesis and validation

The feasible operating procedures of each stage can be produced by synchronizing all component models and the corresponding control specifications. A sequential function chart (SFC) is used in this work to formally summarize the overall operating procedure by piecing together the sequential steps of all stages. A total of four SFCs have been generated in the flash startup example. Specifically,

- SFC-1: (1) Raise the liquid level to $1.5\ \mathrm{m}$; (2) Raise the temperature and level to their set points (i.e., $75℃$ and $2.5\ \mathrm{m}$) in one step; (3) Set controllers on AUTO and maintain stable operation.
- SFC-2: (1) Raise the liquid level to $1.5\ \mathrm{m}$; (2) Raise the temperature to $40℃$ and then to $75℃$ in two steps and raise the level to $2.5\ \mathrm{m}$ in one step; (3) Set controllers on AUTO and maintain stable operation.
- SFC-3: (1) Raise the liquid level to the set point at $2.5\ \mathrm{m}$; (2) Raise the temperature to $75℃$ in one step; (3) Set controllers on AUTO and maintain stable operation.
- SFC-4: (1) Raise the liquid level to the set point at $2.5\ \mathrm{m}$; (2) Raise the temperature to $40℃$ and then to $75℃$ in two steps; (3) Set controllers on AUTO and maintain stable operation.

All four scenarios were tested in simulation studies with ASPEN PLUS DYNAMICS. It was determined that the last two are unsafe in practical applications since the liquid level exceeded the height of flash drum ($5\ \mathrm{m}$) at the end of simulation runs. On the other hand, SFC-1 and SFC-2 are compared on the basis of three performance indices, i.e., the total amount of off-spec products, the total amount of energy consumed and the terminations time (see Table 1). It can be observed that SFC-1 outperforms SFC-2 in every aspect.

Table 1: Comparison of performance indices of SFC-1 and SFC-2.

SFC no.	Total amount of off-spec products (kg)	Total amount of energy consumed (MMkcal)	Termination time (hr)
SFC-1	918.1	10.1	0.90
SFC-2	937.8	10.4	1.05

Let's next take a closer look at the best operating procedure obtained in the flash startup example, i.e., SFC-1 in Figure 10. The initial settings of PID controllers and actuators in this procedure are implemented according to S_0. The first activation conditions in SFC-1 are basically the sensor readings specified in AC_1. After AC_1 is confirmed, valve Vin is supposed to be opened fully as required in S_1. Consequently, the liquid level in flash drum will rise quickly. Upon observing the level reading of $1.5\ \mathrm{m}$, i.e., AC_2, utility heating should be applied and, at the same time, the inlet and outlet flows also begin via the operation steps specified in S_2. The subsequent activation conditions in AC_3 are essentially sensor readings that equal to the set points of temperature ($75℃$) and level ($2.5\ \mathrm{m}$) for the continuous flash operation at steady state. The final steps of the startup operation, i.e., S_3, are operator (or PLC) actions to switch all PID controllers to AUTO modes and adjust their set points to the intended steady-state values respectively. The corresponding ASPEN simulation results are presented in Figure 11. It can be clearly verified that the concentration spec ($87\ \mathrm{wt\%}$ methanol) of the top product is

reached at around 0.9 hr (see the lower-right diagram). It can also be confirmed that the liquid level is far below the height of flash drum (5 m) throughout the entire operation.

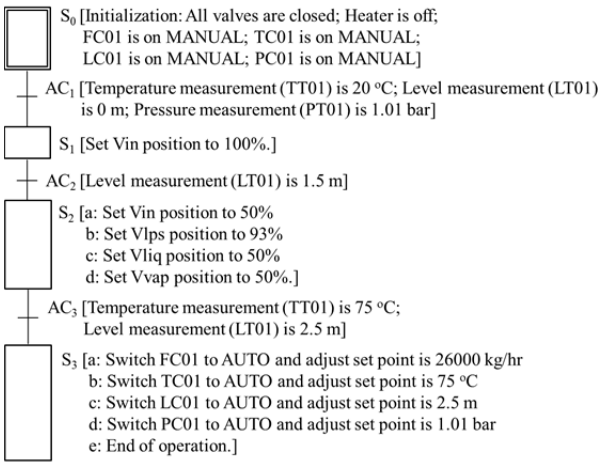

Figure 10: Sequential function chart SFC-1 in the flash startup example.

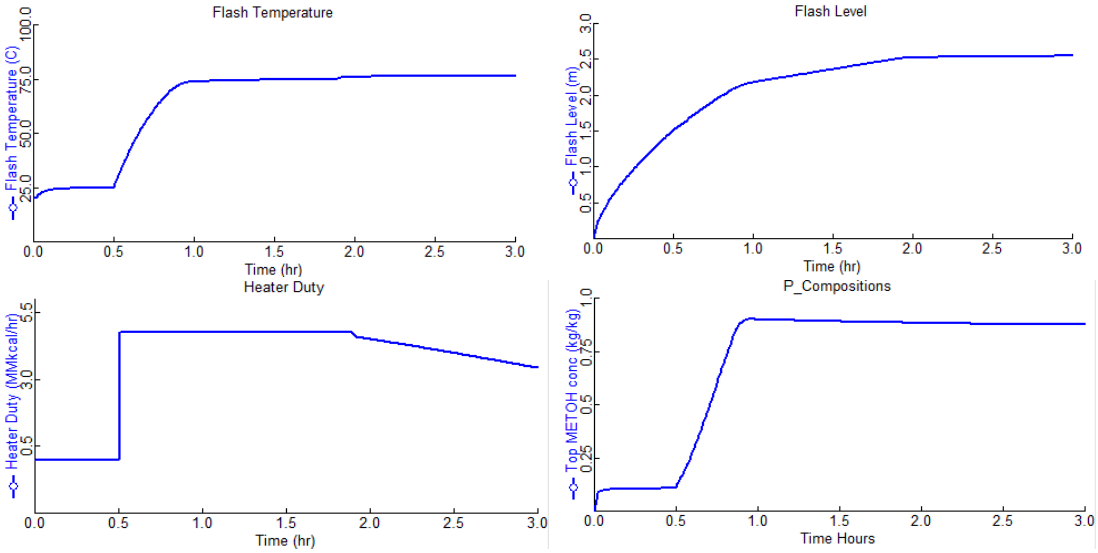

Figure 11: Simulation results of SFC-1 in flash startup operations.

9. Conclusions

A generic approach has been developed in this work for systematically creating operating procedures based on untimed automata. The proposed procedure-synthesis steps include: (1) constructing automaton model for each component in a given PFD; (2) dividing operation into stages and developing automata to represent the control specifications of every stage; (3) assembling the system model of each stage and consolidating them into a single SFC. The commercial Aspen Plus Dynamic was used to validate the candidate SFCs. Finally, it should be emphasized that this approach has also been tested successfully on other more realistic cases, such as the batch reaction process and the distillation startup operation.

References

Åkesson, K., Fabian, M., Malik, R. 2006 SUPREMICA – An integrated environment for verification, synthesis and simulation of discrete event systems. Proceedings of the 8th International Workshop on Discrete Event Systems, IEEE. 384-385.

Li, J.H., Chang, C.T., Jiang, D., 2014, Systematic generation of cyclic operating procedures based on timed automata. Chemical Engineering Research and Design, 92, 139 – 155.

Lu, Y.C., Chen, Z.L., Lee, H.Y., 2017, Optimal start-up strategies for a conventional distillation column using simulated annealing. Chemical Engineering Transactions. 61, 901-906.