Contents lists available at ScienceDirect

Chemical Engineering Research and Design

journal homepage: www.elsevier.com/locate/cherd

Systematic generation of cyclic operating procedures based on timed automata

Jeh-Hsuan Li[a], *Chuei-Tin Chang*[a,*], *Da Jiang*[b]

[a] *Department of Chemical Engineering, National Cheng Kung University, Tainan, 70101, Taiwan, ROC*
[b] *Key Laboratory of Advanced Control and Optimization for Chemical Processes (East China University of Science and Technology), Ministry of Education, Shanghai 200237, PR China*

ABSTRACT

Manual synthesis of cyclic operating procedure in a realistic system is widely regarded as a difficult task since it is both time-consuming and error-prone. It is thus desirable to develop a systematic approach to automatically generate the optimal schedule of operation steps so as to achieve one or more specific production goal. The timed automata are utilized in the present work for such a purpose. In particular, all components in a given system and the corresponding control specifications are characterized with automata according to the proposed modeling rules. By using parallel composition, a system automaton can be produced with these models and the most appropriate operation path can then be identified accordingly. For any practical application, a sequential function chart and the corresponding Gantt chart can also be easily extracted from this path. Three examples are presented in this paper to demonstrate the feasibility of the proposed approach.

Keywords: Timed automata; Modeling method; Cyclic operating procedure

1. Introduction

The operating procedure of a batch chemical process should be synthesized on the basis of the initial system condition(s) and also the ultimate operational goal(s). Its detailed steps are usually represented with a sequential function chart (SFC). Traditionally, the SFCs have been produced with an ad hoc manual approach which often becomes unmanageable as the process complexity increases. To overcome the difficulties caused by combinatorial explosion, there have been many published studies on systematic procedure synthesis. The original problem formulation was first proposed by Rivas and Rudd (1974), and extensive works concerning the design and verification of procedural controllers were then carried out in the later years. The related issues were basically addressed with various modeling/reasoning tools, e.g., the mathematical programming models (Crooks and Macchietto, 1992; Li et al., 1997), the symbolic model verifiers (Yang et al., 2001), the AI-based strategies (Foulkes et al., 1988), and other qualitative models such as Petri nets (Lai et al., 2007; Lee et al., 2011) and the untimed automata (Yeh and Chang, 2012a,b).

Although interesting results were presented in the aforementioned papers, the available methods are still not mature enough for realistic cyclic operations. In particular, every existing method was developed on the basis of a single pre-determined initial condition for batch operation. This approach may not be feasible in the present application if the given system could start at a different (and probably abnormal) state. In addition, the elapsed time of each operation step has not been considered rigorously in the previous work either. To characterize the periodic operating procedure unambiguously, it is desired to produce not only a SFC but also the corresponding Gantt chart to stipulate the time schedule for implementing the operation steps.

To address the aforementioned issues, an improved modeling strategy is developed in this work to build timed automata for characterizing components and specifications in all possible scenarios. A versatile system model can then be

* Corresponding author. Tel.: +886 6 275 7575x62663; fax: +886 6 234 4496.
E-mail addresses: ctchang@mail.ncku.edu.tw, chueitinchang@msn.com (C.-T. Chang).
Received 18 April 2013; Received in revised form 25 June 2013; Accepted 29 June 2013

http://dx.doi.org/10.1016/j.cherd.2013.06.032

synthesized accordingly by applying the standard operation of parallel composition. The best operation path embedded in this model is identified with an existing software, i.e., UPPAAL (Behrmann et al., 2006), and the corresponding operating procedure can also be easily generated. Three examples are presented at the end of this paper to facilitate clear explanation of the proposed method.

The remaining paper is organized as follows. The general framework of automata-based procedure-generation strategy is first described in the following section. A unified hierarchical structure of the batch processes is then presented in Section 3. The systematic methods for constructing timed automata that model the components and also the control specifications are outlined in the next two sections. The desired cyclic operating steps can be identified by combining these automata according to the rules of parallel composition. This procedure-synthesis approach is illustrated with a simple example in Section 6. To demonstrate the effectiveness of the proposed strategy, additional case studies have also been carried out and two of them are presented in Section 7. Finally, the concluding comments are given in the last section.

2. Automata-based procedure-generation strategy

A timed automaton is a finite-state machine equipped with one or more clock (Alur and Dill, 1994). All clocks progress synchronously, and every one of them is described with a dense-time model in which the clock variable assumes a real positive value. To facilitate clear description of the proposed method, a brief summary of the automaton structure is given below. In particular, a timed automaton can be regarded as a six-tuple:

$$TA = (L, \ell_0, C, A, I, E) \quad (1)$$

where, L is a set of locations; $\ell_0 \in L$ is the initial location; C denotes the set of clocks; A is a set of actions. In addition, $I : L \to B(C)$ denotes a function $I(l) = b(c)$ which assigns invariants to locations. Note that $B(C)$ is the set of conjunctions over simple conditions of the form:

$$\{x \oplus c\} \text{ or } \{x - y \oplus c\} \quad (2)$$

where, $x, y \in C$, $c \in \mathbb{N}$ and $\oplus \in \{<, \leq, =, \geq, >\}$. Finally, the set $E \subseteq L \times A \times B(C) \times 2^C \times L$ contains all edges in the automaton. Each edge represents a transition process from one location to another, which is enabled by an action in the set A, constrained by a guard in the set $B(C)$ and timed according to a collection of clocks which belongs to the power set of C, i.e., 2^C.

The default verification tool in UPPAAL is used in the study to search for the best operation path within the real-time system (Pettersson, 1999). UPPAAL is an integrated tool environment for modeling, validation and verification of real-time discrete-event systems (Bengtsson and Yi, 2004; Kim et al., 2006). Although other alternatives, e.g., KRONOS (Bozga et al., 1998) and RED (Wang, 2001), are available, this software is adopted simply for its effectiveness and user-friendliness. More specifically, the optimal operating procedure is produced in four distinct steps in the present study:

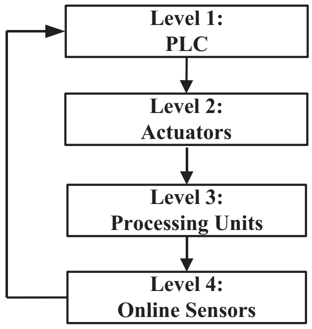

Fig. 1 – Hierarchical structure of a batch process.

(1) Build the automaton models of all components in the uncontrolled plant;
(2) Construct automata to represent the control specifications in all possible scenarios;
(3) Combine all automata created in the above two steps with parallel composition;
(4) Execute suitable property verification function in UPPAAL so as to locate the best operation pathway.

3. Hierarchical structure of batch processes

The hardware items in any batch process can all be depicted in a process flow diagram (PFD). They are treated in this study as components of the given system and classified into a 4-level hierarchy (see Fig. 1). The top-level component is usually a programmable logic controller (PLC) used for implementing a set of pre-determined actions to alter the actuator states in the second level. More than one actuator, e.g., solenoid valves, pumps, compressors, and switches, etc., may be used for adjusting the material and/or energy flow patterns in the given system. Every processing unit in PFD, such as the heat exchanger, separator, reactor and storage tank, is considered as a level-3 component, while every on-line sensor is treated as a component in level 4. The PFD of an uncontrolled batch process, i.e., levels 2 to 4, is assumed to be given in this work, while the SFC and the corresponding Gantt chart are not available. Let us use the simple liquid heating system given in Fig. 2

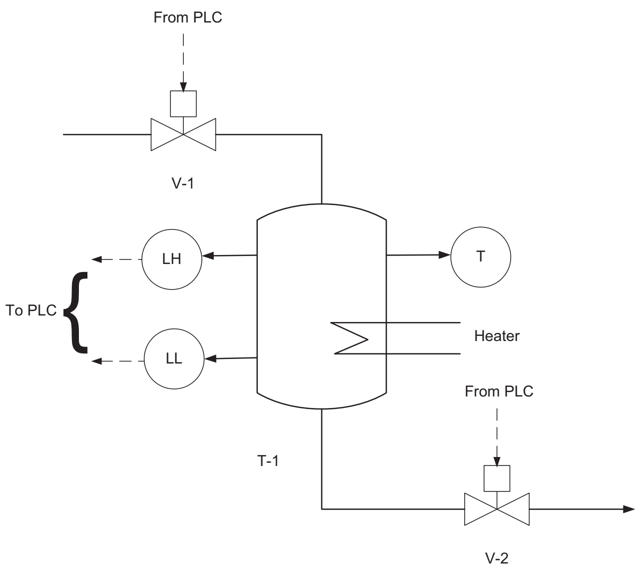

Fig. 2 – PFD of a liquid heating system (Example 1).