Computers and Chemical Engineering 84 (2016) 12–27

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Timed-automata based method for synthesizing diagnostic tests in batch processes

Wei-Chun Hsieh, Chuei-Tin Chang*

Department of Chemical Engineering National Cheng Kung University, Tainan 70101, Taiwan, ROC

ARTICLE INFO

Article history:
Received 15 March 2015
Received in revised form 15 June 2015
Accepted 12 August 2015
Available online 20 August 2015

Keywords:
Timed automata
Diagnostic test plans
Model-checking tools
Batch processes

ABSTRACT

Hardware failures are inevitable but random events in the useful life of any batch chemical plant. If these incidents are not efficiently diagnosed, the consequences can be very serious. In general, two design measures may be implemented *offline* to enhance the overall diagnostic performance, i.e., installing sensors and/or stipulating test plans for *online* implementations. Since the former has already been studied extensively, the present study focuses only upon the latter. In a recent work, Kang and Chang (2014) proposed an effective method to conjecture diagnostic tests using the untimed automata. However, due to a lack of time-tracking mechanisms, the failure-induced behaviours cannot always be characterized adequately with such models. A systematic procedure-synthesis strategy is therefore developed in the present study by making use of the *timed* automata and the model-checking capabilities of existing software, e.g., UPPAAL (Behrmann et al., 2006). All component models are first constructed, and all possible fault propagation scenarios and their observable event traces (OETs) are next enumerated exhaustively. The optimal test plan for every OET can then be established by generating the supervisory controller to improve diagnostic resolution. Extensive case studies have also been carried out in this work to confirm the validity and effectiveness of the proposed approach.

1. Introduction

A large number of high value-added chemical products, such as the specialty chemicals, foods, semiconductors, pharmaceuticals, etc., are often manufactured in complex but flexible batch processes. Hardware failures are random but inevitable events over the lifespan of any such plant. If the root causes of a failure-induced event sequence cannot be correctly identified in time, the final consequences may be catastrophic. Generally speaking, the overall performance of a diagnostic system can be improved by capturing more online data. To this end, the obvious design strategy is to install additional sensors. However, since new hardware inevitably requires extra spending and, also, the related issues have already been discussed extensively in the literature, there are incentives to develop an alternative means for enhancing diagnostic resolution without capital investment. Yeh and Chang (2011) proposed to implement online test procedures for such a purpose, while Kang and Chang (2014) later developed an effective procedure-synthesis method to conjecture the diagnostic tests according to untimed automata.

It should be noted that several studies have already been performed to address various issues concerning fault diagnosis in batch processes. Nomikos and MacGregor (1994, 1995) utilized the multi-way principal component analysis for batch process monitoring, which has later been extended for online diagnosis applications (Kourti and Macgregor, 1995; Kourti et al., 1995; Undey et al., 2003; Lee et al., 2004). Other fault identification tools, such as the artificial immune systems, artificial neural networks and knowledge-based expert systems (Dai and Zhao, 2011; Ghosh and Srinivasan, 2011; Tan et al., 2012; Zhao, 2014), have also been used for diagnosing the batch plants. Although satisfactory results were reported, the above methods are mostly effective for fault diagnosis in systems with relatively few interconnected units and, also, the diagnostic resolution in cases of coexisting failures may not always be acceptable.

In order to expand the scope of diagnosis in realistic applications, Chen et al. (2010) developed several Petri-net based algorithms to configure fault identification systems for plants with many more units. Since the event sequences (or traces) in multi-failure scenarios cannot be conveniently generated with the Petri-net models, their approach was limited to the single-failure incidents. On the other hand, it was found that this shortcoming can in general be avoided with the untimed automata (Sampath et al., 1995, 1996, 1998; Baroni et al., 1999, 2000; Debouk et al., 2000; Benveniste et al., 2003; Zad et al., 2003; Qiu and Kumar,

* Corresponding author.
E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

http://dx.doi.org/10.1016/j.compchemeng.2015.08.007
0098-1354/

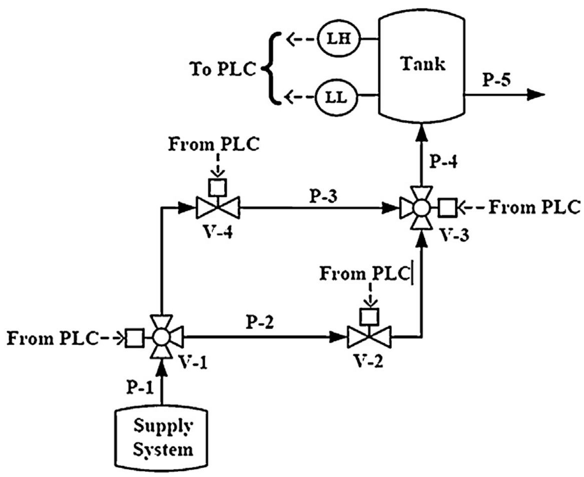

Fig. 1. P&ID of the liquid transfer system in Example 1.

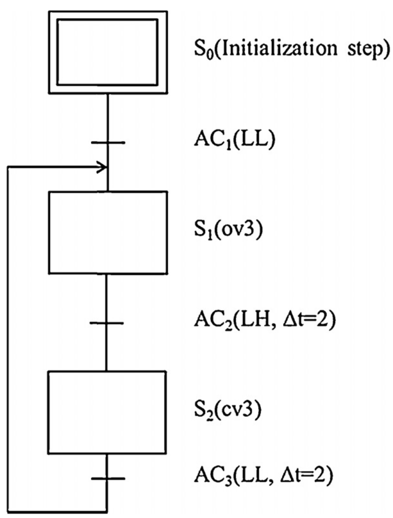

Fig. 2. Normal SFC of liquid transfer operation in Example 1.

2006; Yeh and Chang, 2011). A so-called "diagnoser" can be constructed accordingly to predict all observable fault-propagation event sequences (or "traces") and to pinpoint the corresponding fault origins. In a later study, Gascard and Simeu-Abazi (2013) improved this approach by using the timed automata to build diagnosers for the dynamic discrete-event systems.

Since the root cause(s) of a trace in the diagnoser may or may not be unique, it is desirable to further enhance the diagnostic resolution with other nonconventional means. As mentioned previously, Kang and Chang (2014) have developed a systematic method to generate the test plans for upgrading a given diagnoser without capital investment. However, due to the lack of time-tracking mechanisms in their *untimed* models, the failure-induced behaviours cannot always be characterized adequately. To overcome this drawback, it is obviously reasonable to make use of the *timed* automata for the purpose of generating more comprehensive plans. Notice that such models have already been utilized to address other closely related issues. For examples, they were used to verify if any given procedure conforms to the design specifications (Lohmann et al., 2006; Kim and Moon, 2009, 2011; Lahtinen et al., 2012), and Li et al. (2014) also proposed a systematic approach to synthesize controller actions for periodic operations.

Finally, to facilitate clear illustration of the proposed approach, the general procedure for test-plan synthesis is summarized in the sequel:

1. All embedded components in the given process are first modelled with the timed automata.
2. All possible fault propagation scenarios and their observable event traces (OETs) are next enumerated exhaustively.
3. The optimal test plan for every OET is then established by generating the supervisory controller to achieve a higher degree of diagnostic resolution.

The resulting test plans can then be implemented online after observing any of the OETs in diagnoser during actual operation.

2. General approach to build plant model

Since the modelling principles proposed by Kang and Chang (2014) are generic enough, their basic approach is adopted to build time automata in the present study. For the sake of clarity, this model construction method is illustrated here with a simple example. Specifically, let us consider a fictitious liquid transfer system represented by the piping and instrumentation diagram (P&ID) in Fig. 1 and also the sequential function chart (SFC) in Fig. 2. Notice that the components in this and any other batch process can be classified into a hierarchy of 4 different levels: (1) the programmable logic controller (PLC); (2) the actuators, i.e., the three-way valves (V-1 and V-3) and the two-way valves (V-2 and V-4); (3) the processing units, i.e., the buffer tank' and (4) the online sensor(s). If a three-way valve is closed, the port connecting to the horizontal pipeline in Fig. 1, i.e., pipe P-2 in the case of V-1 or pipe P-3 in the case of V-3, is assumed to be blocked. Otherwise, its inlet flow(s)

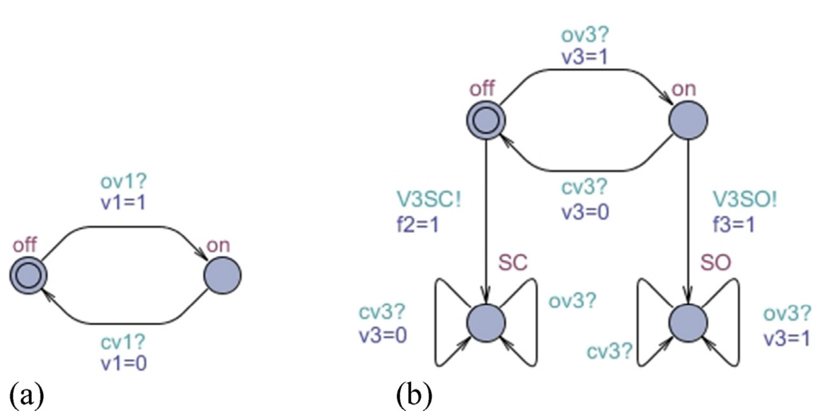

Fig. 3. Valve models for (a) V-1 and (b) V-3 in Example 1.