Automatica 50 (2014) 915–923

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Robust control reconfiguration of resource allocation systems with Petri nets and integer programming[☆]

Jun Li [a,b], MengChu Zhou [c,d,1], Tao Guo [a,b], Yahui Gan [a,b], Xianzhong Dai [a,b]

[a] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, China

[b] School of Automation, Southeast University, Nanjing 210096, China

[c] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China

[d] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Article history:
Received 25 September 2012
Received in revised form
18 October 2013
Accepted 10 November 2013
Available online 3 January 2014

Keywords:
Discrete event systems
Supervisory control
Reconfiguration
Petri nets
Integer programming

ABSTRACT

Supervisory control reconfiguration can handle the uncertainties including resource failures and task changes in discrete event systems. It was not addressed to exploit the robustness of closed-loop systems to accommodate some uncertainties in the prior studies. Such exploitation can cost-efficiently achieve reconfigurability and flexibility for real systems. This paper presents a robust reconfiguration method based on Petri nets (PNs) and integer programming for supervisory control of resource allocation systems (RASs) subject to varying resource allocation relationships. An allocation relationship is seen as a control specification while the execution processes requiring resources as an uncontrolled plant. First, a robust reconfiguration mechanism is proposed. It includes updating the P-invariant-based supervisor and evolving the state of the closed-loop system. The latter adapts to the control specification changes by the self-regulation of the closed-loop system's state. Next, two novel integer programming models for control reconfiguration are proposed, called a reconfiguration model with acceptability and reconfiguration one with specification correction. Since both models integrate the firability condition of transitions, no additional efforts are required for the state reachability analysis. Finally, a hospital emergency service system is used as an example to illustrate them.

1. Introduction

Most of today's production and service systems are man-made discrete event systems (DESs). From the perspective of DES supervisory control (Basile & Chiacchio, 2007; Charbonnier, Alla, & David, 1995; Ramadge & Wonham, 1987), *tasks* can be seen as the *control specifications*, or *specification* for short, while systems that will handle them as *uncontrolled plants*. According to a specification, a supervisory controller is designed and then acts on the plant to make it behave as desired. With the growing trend towards the small-lot and customized need, production systems are required to possess sufficient flexibility and reconfigurability to adapt to frequent changes in their tasks and to smoothly switch their production. To achieve this goal, their physical entities are designed to be modular, flexible and reconfigurable. Based on such design, the superiority compared with the traditional way can be achieved if the control part can adapt to or rapidly respond to changes in the specification by modifying its control logic and structure. It is a cost-efficient approach to product customization. The corresponding process taking place in a DES is referred to as *supervisory control reconfiguration* (Li, Dai, & Meng, 2009; Li, Dai, Meng, & Dou, 2009; Li, Wu, & Zhou, 2012; Li, Zhou, & Dai, 2012; Liu & Darabi, 2004; Sampath, Darabi, Buy, & Jing, 2008). Its applications include flexible manufacturing systems (FMSs) (Zhou & DiCesare, 1993; Zhou & Venkatesh, 1998), reconfigurable manufacturing systems (Koren et al., 1999), web service composition (Tan, Fan, & Zhou, 2009; Tan, Fan, Zhou, & Tian, 2010; Xiong, Fan, & Zhou, 2009), and workflow (Tan & Zhou, 2013).

At present, supervisory control reconfiguration is a hot topic in DES. Its origin can go back to the concept of *reconfigurable DESs* whose states and structures both change over time (Garcia & Ray, 1996). Early work deals with the fault-tolerance and recovery. Darabi, Jafari, and Buczak (2003) propose a technique

☆ This work was supported in part by the National Natural Science Foundation of China under Grants 61374069, 61374148, 61004035, 61175113, 51105076, and 51005160 and Doctoral Programs Foundation of Ministry of Education of China under Grant 20130092110022. The material in this paper was partially presented at the 9th IEEE International Conference on Automation Science and Engineering (IEEE CASE 2013), August 17–21, 2013, Madison, Wisconsin, USA. This paper was recommended for publication in revised form by Associate Editor Bart De Schutter under the direction of Editor Ian R. Petersen.

E-mail addresses: j.li@seu.edu.cn (J. Li), mengchu@gmail.com (M.C. Zhou), taoguo168@gmail.com (T. Guo), ganyahui@163.com (Y. Gan), xzdai@seu.edu.cn (X. Dai).

[1] Tel.: +1 973 596 6282; fax: +1 973 596 5680.

0005-1098/$ – see front matter © 2014 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.automatica.2013.12.015

called observable projection to perform the dynamic reconfiguration of a DES controller in response to the dynamic changes in the observation set. On this basis, Liu and Darabi (2004) design an independent control reconfiguration mechanism by the finite time observation principle and feedback adjustment strategy. Their controller can achieve desired permissiveness. To treat the failure of control channels, Iordache and Antsaklis (2004) achieve fault-tolerance by reconfiguring PN-based supervision with uncontrollable transitions and provide several reconfiguration policies for a closed PN system in response to changes in the specification. Their work is an important step towards supervisory control reconfiguration. Haji and Darabi (2007) introduce a reconfiguration method for project management systems using PNs. It transforms the reconfiguration responses of supervisory control systems to the changes in both project task and resource information into the transition firing sequence. A firing sequence is found via integer programming. Based on such work, Sampath et al. (2008) take a hospital service system as an example to present an optimal reconfiguration method of PN controllers subject to changing resources. These efforts have contributed to control reconfiguration and opened new research areas. However, the reconfiguration area is still in its infancy with the following problems:

(1) The robustness of closed-loop systems in response to some specification changes is ignored. Usually, a closed-loop system is able to directly respond to some changes through self-regulation. In the existing methods, such an ability has not been paid attention and modifying the specification is their definite option for reconfiguration; and
(2) Reconfiguration faces the state reachability problem. It is NP-hard and has not been solved well.

Both problems motivate this work. Reconfigurable systems to be studied belong to resource allocation systems (RASs) (Reveliotis, 2005). An RAS is a typical DES where many processes compete for limited resources. Its resource allocation relationship can describe a given task. It is required to respond rapidly to task changes. However, little attention is paid to how to make a supervisor respond to a changing task without causing trouble. This paper presents a method for control reconfiguration of RASs with varying tasks:

(1) A robust control reconfiguration principle and mechanism is given to include the acceptability analysis of specifications. We have revealed the robustness and systematically studied the reconfiguration of closed-loop systems to accommodate different specification changes for the first time in the field of DES modeled with PN. Three types of changes are fully considered, i.e., compatible, incompatible but acceptable, and unacceptable. Our mechanism avoids the shortcomings in Haji and Darabi (2007) and Sampath et al. (2008) where the robustness of the closed-loop system cannot be estimated and fully used, thereby increasing the reconfiguration cost due to unnecessary modification of the specification.
(2) Two integer programming models for control reconfiguration with robustness analysis are proposed. They are used to find a firing sequence of transitions as a serial of reconfiguration actions to achieve the state evolution of a closed-loop system. To implement the robust reconfiguration, it requires that one be used prior to another and the latter is used only if the new specification is unacceptable. Since both models integrate the firability condition of transitions, no additional efforts are required for the state reachability analysis as required by the existing work, e.g., Sampath et al. (2008). Also, this work considers for the first time the transition firing cost and firing step length of concurrent transitions as a single optimization objective of reconfiguration.
(3) Taking a hospital emergency service system as an example, we use three scenarios to show the proposed method.

The preliminary is recalled in Section 2. Section 3 presents the reconfiguration mechanism. Section 4 proposes and compares the two models. Section 5 presents their application. The last section concludes the paper.

2. *P*-invariant-based supervisor design

Suppose that the system to be controlled is represented by a PN $N = (P, T, F, W)$, called a plant, where $P = \{p_1, p_2, \ldots, p_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ are the sets of places and transitions with $P \cap T = \varnothing$, $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, and $W : F \to \mathbb{N}$ is the weight function where $\mathbb{N} = \{0\} \cup \mathbb{Z}^+$ is the natural number set and $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$. $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$. A node $x \in P \cup T$, ${}^{\cdot}x = \{y \in P \cup T | (y, x) \in F\}$ is called the preset of x, while $x^{\cdot} = \{y \in P \cup T | (x, y) \in F\}$ is called the postset of x. The incidence matrix of a PN is a matrix $D : P \times T \to \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers, such that $D(p, t) = W(t, p) - W(p, t)$. It is partitioned into two matrices, i.e., pre-incidence one D^- and post-incidence one D^+, where $D^+(p, t) = W(t, p)$ while $D^-(p, t) = W(p, t)$.

A marking vector μ is a mapping from P to $\mathbb{Z}$. $t \in T$ is called *enabled* at μ, denoted by $\mu[t\rangle$, if $\forall p \in {}^{\cdot}t, \mu(p) \geq W(p, t)$. It can *fire*, resulting in a new marking μ', denoted by $\mu[t\rangle\mu'$, where $\mu'(p) = \mu(p) + D(p, t), \forall p \in P$. In this case μ' is *reachable* from μ. Generally, μ' is *reachable* from μ if there is a firing sequence of transitions $\pi = t_1 t_2 \ldots t_k$, $t_i \in T$ and $i \in \mathbb{Z}_k^+ = \{1, 2, \ldots, k\}$ such that $\mu[\pi\rangle\mu'$. Let $R(N, \mu_0)$ denote the *reachability set* of net N, i.e., all reachable markings of N from the initial marking μ_0. A distinguished property of a PN is its ability to describe *concurrency* that can be reflected by multi-transition firings in a single *step*. It means that more than one enabled transition can fire in a step. The transitions that fire in the same step are called *concurrent transitions*. Let vector X_i record the number of firing times of every transition in firing step i. If μ' is *reachable* from μ, there is a sequence of firing steps $\pi = \theta_1\theta_2 \cdots \theta_k$, where θ_i is the *concurrent (simultaneous firing) transition set* in firing step $i \in \mathbb{Z}_k^+$, such that $\mu[\pi\rangle\mu'$ and $\mu' = \mu + D \cdot \left(\sum_{i=1}^{k} X_i\right)$. Given a firing sequence $\pi = \theta_1\theta_2\theta_3$ with $\theta_1 = t_1 t_5$, $\theta_2 = t_7 t_3 t_2$, and $\theta_3 = t_8 t_8 t_9$, we write $\pi = \{t_1 t_5\}\{t_7 t_3 t_2\}\{t_8 t_8 t_9\}$ but not $t_1 t_5 t_7 t_3 t_2 t_8 t_8 t_9$.

The marking of a plant net can satisfy linear inequality constraints, called a *control specification* or *specification* for short:

$$L\mu \leq B \tag{1}$$

where $L \in \mathbb{Z}^{m_c \times m}$ and $B \in \mathbb{Z}^{m_c}$, in which $\mathbb{Z}$ is the set of integers, m_c is the number of constraints, and $m = |P|$. The specification in (1) is powerful enough to adapt to most of the applications where the state, quantity, and logic relationships exist (Yamalidou, Moody, & Antsaklis, 1996).

Specification (1) can be transferred to the form of linear equations by adding a relaxation term. If each newly added term is seen as the marking of a new place, a *P-invariant* is formed. In it, the weighted markings of all the involved places do not change with the state evolution. Based on *P*-invariants, a controller enforcing (1) can be designed, consisting of the new places called *control places*. This is the design principle of a *P*-invariant based supervisor proposed by Yamalidou et al. (1996). The net composed of control places, their input and output arcs, and their linked transitions in a plant net is called a supervisor. Suppose that D and μ_0 are the incidence matrix and initial marking of the plant net. The incidence matrix D_s and initial marking $\mu_{0,s}$ of the supervisor are:

$$D_s = -L \cdot D \tag{2}$$

$$\mu_{0,s} = B - L \cdot \mu_0. \tag{3}$$