The Journal of Systems and Software 83 (2010) 675–688

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Petri net modeling and deadlock analysis of parallel manufacturing processes with shared-resources

Farooq Ahmad *, Hejiao Huang, Xiao-Long Wang

Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, PR China

ARTICLE INFO

Article history:
Received 3 February 2009
Received in revised form 9 November 2009
Accepted 9 November 2009
Available online 13 November 2009

Keywords:
Petri net
Transition vectors
Deadlock
Parallel processing flows

ABSTRACT

Multiple resource-sharing is a common situation in parallel and complex manufacturing processes and may lead to deadlock states. To alleviate this issue, this paper presents the method of modeling parallel processing flows, sharing limited number of resources, in flexible manufacturing systems (FMSs). A new class of Petri net called parallel process net with resources (PPNRs) is introduced for modeling such FMSs. PPNRs have the capacity to model the more complex resource-sharing among parallel manufacturing processes. Furthermore, this paper presents the simple diagnostic and remedial procedures for deadlocks in PPNRs. The proposed technique for deadlock detection and recovery is based on transition vectors which have the power of determining the structural aspects as well as the process flow condition in PPNRs. Moreover, the proposed technique for dealing with deadlocks is not a siphon-based thus the large-scale PPNRs for real-life FMSs can be tackled. Finally, the proposed method of modeling and deadlock analysis in the FMS having parallel processing is demonstrated by a practical example.

1. Introduction

The complex nature of flexible manufacturing systems (FMSs) having interacting parallel processes requires its modeling and analysis for the optimal operational planning and policy. Handling of shared-resources becomes an important aspect during FMS design and allocation of resources is always a challenging task in the modeling of FMS. Further, the resource-sharing among different operations to be performed does not imply its simultaneous usage because each operation utilizes a resource type exclusively and releases it after completion. Moreover, a resource type held by one operation cannot be preempted by another operation. Therefore, parallel flow of multiple jobs in a FMS with limited number of resources may lead to a deadlock situation. For instance, parallel or concurrent processing of raw products of different types on limited number of resources such as machines, robots, buffers etc. is a common situation in FMSs. The parallel flows of multiple products in a FMS can be identified as different jobs to be completed. While sharing the limited number of resources, these jobs have a particular operation routing that determines the order in which resources must be assigned to the product. Therefore, the allocation of limited number of resources to different operations to achieve the efficient output of a FMS is not a trivial task and requires the modeling of its operation flows.

The modeling of FMSs can be based on a number of different formal methods, such as Petri nets (PNs), formal languages generated by finite automata and directed graphs. The formalism of finite automata for the modeling of FMSs has been adopted by several researchers (Ramadge and Wonham, 1987; Reveliotis and Ferreira, 1996; Yalcin and Boucher, 2000) while the graph-theoretic approach has also been used as a modeling tool for FMSs (Cho et al., 1995; Fanti et al., 1996). Among these, PNs are a widely used formalism to model various aspects and characteristics of FMS operation flows in systematic way.

There are significant advantages of PNs with respect to other formalisms due to the following reasons. First, a PN state is a vector of non-negative integers, where the state space for automata is a symbolic unstructured set. Further, many analysis techniques (e.g., linear-algebraic techniques, net reduction and refinement methods, etc.) have been developed for PN based models which do not require the state space enumeration and related computation can be made by utilizing their structural information. Second, PNs are suitable for describing the precedence relationships that characterize the order of resource allocation for processing the jobs in FMSs. In addition, PNs are capable of describing the sequence of events that explain the dynamics of FMSs. Third, PNs have the capacity to represent graphically and visualize the primitives such as parallelism, concurrency, conflicts, mutual exclusion, etc. and each of these primitives corresponds to a clear PN structure. Whereas, automata can only describe the interleaving of events and not their simultaneous occurrence thus concurrency may not be represented through automata. PN models for FMSs also have

* Corresponding author. Tel.: +92 3324655020.
E-mail addresses: farooq190@hotmail.com (F. Ahmad), hjhuang@hitsz.edu.cn (H. Huang), wangxl@insun.hit.edu.cn (X.-L. Wang).

0164-1212/$ - see front matter
doi:10.1016/j.jss.2009.11.705

advantages over their digraph models. In the digraph model of a FMS, vertices only represent the system resources. Whereas, PNs are bipartite digraphs whose places-type vertices represent the available and used resources and transitions describe the events representing changes in resource allocation (Fanti et al., 2000).

PNs have been recognized as a powerful tool for modeling FMSs (Ezpeleta and Martinez, 1993; Kamath and Viswanadham, 1986; Sun et al., 1994; Zhou et al., 1992; Zhou and Venkatesh, 1999) because of their ability of detecting the flaws such as deadlocks and capacity overflows in the modeled systems. The PN model of a FMS is either constructed by a top-down or bottom-up approach (Huang, 2004). In a top-down approach (Hamerrlian, 2003; Lee et al., 1987; Zhou and DiCesare, 1991), first the high-level description of the system is presented and then the model is stepwise refined until a complete net model is achieved. For refinements, subnets are added to original high-level net and desired properties are preserved in resultant expanded net. For FMSs with shared-resource, the top-down approach has a drawback that the subnets are strongly coupled and it is hard to find small size of final expanded model.

On the other hand, in bottom-up approach (Ezpeleta et al., 1995; Huang et al., 2003; Souissi, 1990), the net modules of specified subsystems are constructed and finally they are combined by sharing common places, transitions or subnets. But the verification of desired set of properties for such type of integrated model is not straightforward. Therefore, the disadvantage of bottom-up approach is the great difficulty in the verification of desired behavior of the large-scale combined model.

To cope with these shortcomings of previous modeling methods, this paper presents a new class of PNs called *parallel process net with resources* (PPNRs) for FMSs having parallel processes. There are two stages which comprise the modeling procedure of PPNRs. First the parallel process net (PPN) is constructed to specify the process flows of each part type in a manufacturing system without considering the resources. The step of PPN portraits the parallel processing of parts and depicts the order of different operations. Further, this step assists for the proper assignment of resources required to process each part type and provide the resource-allocation policy according to the given production plan. Thereafter, the marked resource-places denoting the availability of resource types are added to the PPN such that input and output transitions of each operation-place act as its output and input transition, respectively. In this way, PPNRs can model more complex resource-sharing and interacting parallel processes in FMSs.

1.1. Literature relevant to the deadlock problem

One of the basic and fundamental issues regarding the correctness of system model is its smooth execution without any deadlock. Deadlock is an undesirable situation in a resource allocation system. In FMSs, deadlocks often cause unnecessary costs such as long downtime and low utilization of some critical and expensive resources, and may lead to undesired results. The most common condition of deadlock in FMS is "circular waiting" (Banaszak and Krogh, 1990), a situation in which two jobs hold a machine but both of them are waiting for the second machine occupied by the other. Banaszak and Krogh (1990) presented a PN model for concurrent jobs flow and proposed a deadlock avoidance algorithm (DAA). An optimal deadlock prevention policy based on the reachability graph analysis of PN model of a given FMS using the theory of regions has been proposed by Uzam (2002). Zandong and Lee (2005) proposed a deadlock state equation based on restrictive PN controllers for deadlock avoidance. A deadlock-checking approach for one-place unbounded PNs based on modified reachability trees (MRTs) has been presented in Ding et al. (2008). A method based on transitive matrix for deadlock detection in the PN model of job shop scheduling problem has been presented in Li et al. (2008a,b).

Liveness property of the PN model implies its freedom of global or local deadlock situation. Over the last decade, significant research has been conducted in the direction of siphon-based characterization of liveness and liveness-enforcing supervision of the PN models of FMSs (Li and Zhou, 2006; Li et al., 2008b; Uzam and Zhou, 2006, 2007). Moreover, Method of Minimal Siphons has been recognized as a main technology used for deadlock detection and analysis (Chu and Xie, 1997; Hu et al., 2006; Li and wei, 2007; Li and Zhou, 2008). However, the number of siphons grows quickly beyond practical limits and in the worst case grows exponentially fast with respect to the PN size (Boer and Murata, 1994; Cordone et al., 2003) and siphon-based liveness-enforcing approaches become restricted and degraded. In addition, they suffer from the computational complexity problem since it is known that in general the complete siphon enumeration in a PN is NP-complete. Furthermore, siphons usually lead to a much more structurally complex liveness-enforcing PN supervisor than the original PN model.

To tackle this problem, an effective and simple technique, without siphons taking into account, based on transition vectors (TVs) (Ahmad et al., 2008) for deadlock detection in PPNRs is presented. Because the proposed technique for deadlock detection and recovery in not a siphon-based, the large-scale PPNRs of real-life FMSs can be handled, which is the main advantage of the proposed technique. TVs describe the structure of PN model into the vectors in a systematic and simplified manner and explain the relation between places and transitions of a net in such a way that they provide complete information regarding structure. The TVs of PPNRs make explicit the condition of operation flow through shared-resourcess by using the transitions. Therefore, they can be used as a powerful tool for deadlock detection. After deadlock detection, attention is focused in this paper to characterize the deadlock state in PPNRs in order to recover the deadlock-free PN model of a FMS.

TVs have two main advantages with respect to the conventional incidence matrix (Murata, 1989; Peterson, 1981). First, different structural aspects in PN model e.g. self-loops, structural conflicts, synchronizations, etc., can directly be realized through TVs but they cannot be observed through the incidence matrix. The structural identification through TVs is briefly explained in Section 5.1. Second, we can directly observe the token flow relation through TVs, which is used for deadlock detection in PPNRs. Indeed, we cannot directly observe the token flow condition in incidence matrices.

1.2. Outline of the paper

Following the introduction in Section 1, some related terminologies are introduced in Section 2. In Section 3, the TVs and their construction are discussed. In Section 4, the focus is on the modeling of parallel manufacturing processes with resources. The TV-based algorithm for deadlock detection and the procedure for PPNRs model recovery are presented in Section 5. Section 6 presents a demonstration of modeling and deadlock analysis of PPNRs through a real-life example of a FMS. Some concluding remarks are presented in Section 7.

2. Definitions and concepts

In this section, some basic definitions and notations of ordinary (for the sake of simplicity) PN are described. The related terminology and notations are mostly taken from (Murata, 1989; Peterson, 1981).