PROCESS MONITORING FOR FLEXIBLE MANUFACTURING SYSTEMS WITH ERROR RECOVERY

Călin Ciufudean calin@eed.usv.ro
Constantin Filote filote@eed.usv.ro
Adrian Graur adriang@eed.usv.ro

University "Ştefan cel Mare" of Suceava, Faculty of Electrical Engeneering
9 University Str., Suceava, RO-720225, Romania

Abstract: This paper focuses on the process monitoring flexible manufacturing systems dynamics using Petri nets models. This paper enhances a Petri net synthesis theory with the capability to deal with reversibility, a property related to error recovery. An algorithm, for modelling the information of each modular Petri net model in the entire system, and for the integrated process monitoring is presented. The class of Petri nets chosen for modelling shared-resource automates manufacturing systems is conservative and possess structural liveness under some conditions which implies the nets reversibility. Thus, the liveness-checking algorithm can be applied to the reversibility checking. An example illustrates the presented approach.

Key words: Controlled Petri nets, liveness, conservativeness, reversibility.

1. INTRODUCTION

A flexible manufacturing system (FMS) consists of a number of systems, such as process actions, material storage, material processing devices, raw and finite material transportation devices, control units, etc. The material flows among the flexible manufacturing cells, machines and equipment are usually connected through an automated material handling system. Production control units, including process information and control commands are routed via a communication system. The communication system can have computers, control units, local area networks. The FMS can manufacture diverse types of products in variable batch sizes [1], [2]. The flexible route, process, machine, etc., meet fast transition of customer requirements. Therefore, a FMS has the ability to cope with rapid market and demand changes. Obviously, such complex systems demand huge Petri net models, which are very difficult to manage. Therefore the subsystems (e.g., the above mentioned systems which compound a FMS) must be modelled with Petri nets, and they will represent the net modules. Merging the net modules, we build the Petri net model of the FMS. This paper enhances a Petri net synthesis theory with the capability to deal with reversibility [3]. We assume that the reader is familiarized with Petri nets, or we refer the reader to [4], [5].

Petri net reversibility property, which means that the net can return to the initial state from any state, is related to the concept of error recovering manufacturing [6] because in the presence of some error, the system may automatically be reinitialized through a recovery process. For example, in railway sorting operations, two convoys fail to mate. A reversible Petri net model of the sorting operation implies that automatic error recovery is possible. The paper is organized as follows: Section II reviews the definition of net modules and the process of merging modules to build a system model. Section III introduces an algorithm for verifying the net reversibility. Section IV presents an example, from the railway sorting operations, in order to illustrate the given approaches.

II. MERGING NET MODULES OF A FMS

Each module in the Petri net model of a FMS synthesizes the process that control a resource type such as a machine pool, buffers, AGV's in an automated manufacturing system (AMS). These modules are modelled with sure Petri nets (e.g., strongly connected state machine – SCSM) in order to ensure the conservativeness (the Petri net is a bounded net) and liveness of the net. Thus, the reversibility of the Petri net model of the FMS is ensured by the merging mechanism of the net modules. Let G_oPN denote the Petri net model of the FMS. We refine the assumption given in [7]: the G_oPN it is an SCSM with few initials marked places called resource places. The rest of the places in the G_oPN are called operation places. Tokens in a resource place denote resource availability, while tokens in an operation place means that certain operations holding some resources are in process. The interactions or synchronizations among the net modules are modelled as the common transitions and common transition subnets. A common transition is a transition contained in more than one module. A transition subnet is a subnet where all the places have no connections outside the net. A common transition subnet is a transition subnet contained in more than one module. A common transition subnet models a common activity controllable by the net containing it. Two restrictions on merging modules are given as follows [7], [8]:

1) At each common transition, there exists at most one input place that is an operation place. The objective of this restriction is to exclude certain net structures in order to simplify the reversibility cheeking. For example, if two paths are to be merged, then it can be known which path contains the resource that is allocated fist, because the resource places are the only places that are initially marked in the merged net.
2) Common transition subnets should not contain resource places. This restriction is necessarily in order to avoid the bottlenecks in the G_oPN, when allocation of resources to the down-stream nets it seems to be a free-choice problem, and not a SCSM one.

III. REVERSIBILITY CONTROLLER FOR FMS's

Quality control and management systems must be analyzed using collected data, and then the result is used to control the process and prevent defects. When process conditions change, the process parameters must be adjusted according to process variability. Reversibility can be used to investigate and control the process, as shown in Fig.1. Our approach for the reversibility significance may be different from the classical ones, but we see this as a controller for the process.

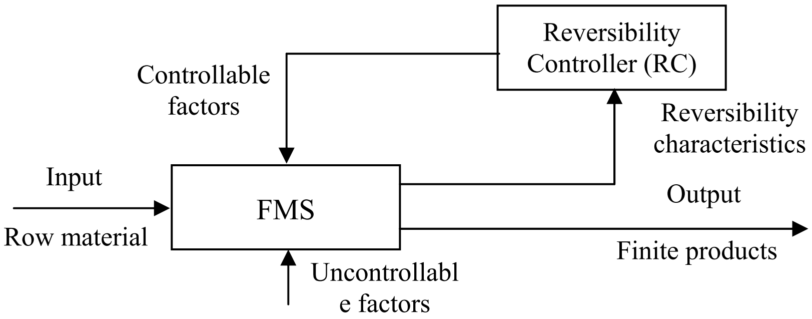

Fig.1. Reversibility controller for a FMS

The reversibility controller (RC) actives in this paper are constructed in terms of G_oPN: the RC actives in Fig.1 can be refined into a G_oPN, as shown in Fig.2. The G_oPN elements are described as follows. RC_P$_1$ represents the data collected from the shop floor. RC_T$_1$ indicates the initial data set, and RC_T$_2$ indicates the subsequent data set. RC_T$_1$ and RC_T$_2$ are controlled and mutually excluded by RC_P$_2$. The number of tokens in RC_P$_2$ indicates the required initial number for analyzing the capability of the process. If a token exists in RC_P$_2$, then RC_T$_2$ is inhibited. RC_T$_6$ represents the capability analysis of the process.

The process capability is used to characterize the process capability with respect to the standard specification of the process. For us, the process capability can be evaluated in terms of the reversibility capability.

RC_P$_5$ contains the data analyzed by the process capability. The marking of RC_P5 controls RC_T3 and RC_T4, which represent the reversible and non-reversible process, respectively. If a process (e.g., a net module) is analyzed as non-reversible, then RC_P$_6$ restarts the capability analysis of the process after adjusting the process parameters.

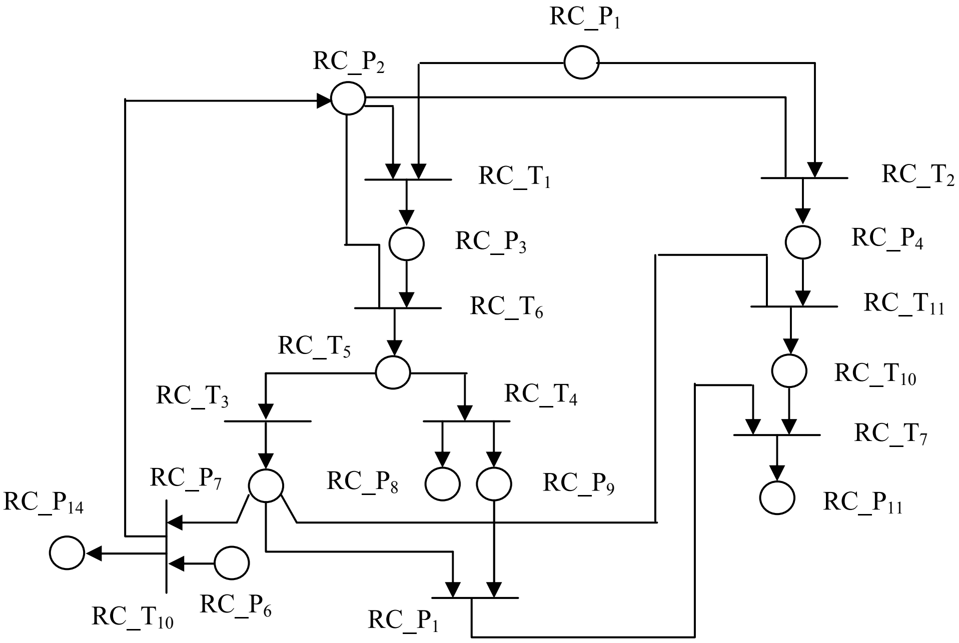

Fig.2. G_oPN model of a RC

If the G_oPN is reversible, then RC_T$_{11}$ is not inhibited and the process is considered to be reversible, where the subsequent location RC_P$_{11}$ stores the results of the control analysis. The inhibitor are connected to RC_T$_2$ is designed for controlling the number of initial data simulations for analysing the process reversibility.

There is a specified number of tokens (depends on the FMS's initial requirements) in RC_P$_2$ in the initial marking. Depending on the token existence in RC_P$_2$, transition RC_T$_2$ is inhibited when a measurement (token) is collected from RC_P$_1$. The inhibitor are connected to RC_T$_6$ implies when the initial data (measurements) are complete, and the process can be executed to generate the reversibility of system. Consequently, the inhibitor are connected to RC_T$_{11}$ indicates if the G_oPN reversibility it is not acceptable, then RC_P$_7$ contains tokens, and hence inhibits the firing of transition RC_T$_7$. The following example of shunting operation in railway system will highlight the above given approach.

IV. AN EXAMPLE OF SHUNTING OPERATION IN RAILWAY SYSTEMS

Considering that the shunting operation is a key activity for increasing the operability of the railway transport systems, we choose this example: a railway system composed from a railway station, a shunting yard, a shunting board, and two railway engines.

We consider the following shunting process: a railway machine takes a wagon convoy C from an industrial railway and binds it to another convoy R brought from another industrial railway. This assembly is formed in a shunting controlled railway, together with another convoy H brought from the kicking horse pass of the shunting yard. This new convoy is trigged (e.g., by the railway engine M_2) at a station railway, in order to be sent. The net representing this assembly operation is given in Fig.3. Each task of the plan P_1 (transitions t_p and t_q are given in Fig.4, and respectively in Fig.5.

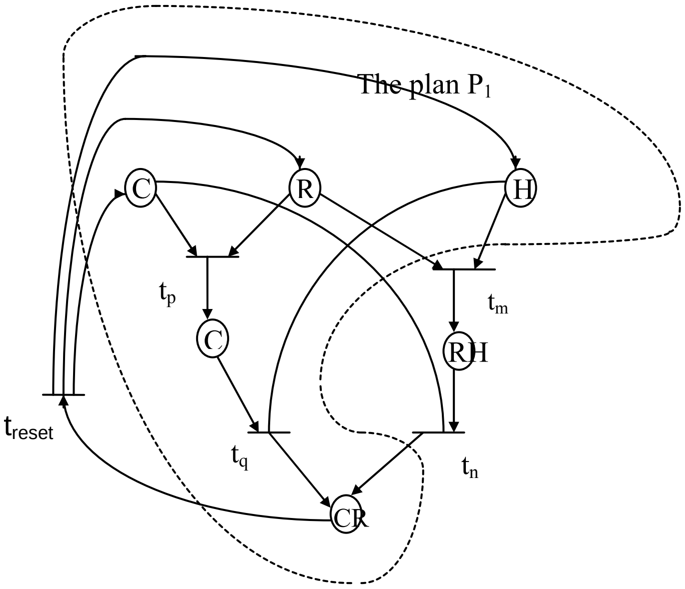

Fig.3 The reperesentation of a shunting operation

The indices in Fig.4 Fig.5, are:

C_{OK} (R_{OK}, H_{OK}) = the convoy C (R or H) is ready to be tugged by a railway engine;
M_{1OK} (M_{2OK}) = the railway engine M_1 (M_2) is available;
M_1C (M_1R, M_1H) = locomotiva M_1(M_2) is attached to the convoy C (R, or H);
M_1M_2CR (M_1M_2CRH) = the railway engine M_1 (M_2) is attached to convoy CR (CRH);
CRH_{OK} = the convoy CRH is ready to go;
M_{1trC} (M_{1trR}, M_{1trH}) = the engine M_1(M_2) tugs the convoy C, R, or H;
CR_{bind} (CRH_{bind}) = the convoys C and R are connected (analogous for convoys C, R şi H).
M_{1elCR} (M_{1elCRH}) = the engine M_1 liberates the convoy CR (or CRH).
exp. = the engine M_2 sends the convoy CRH.

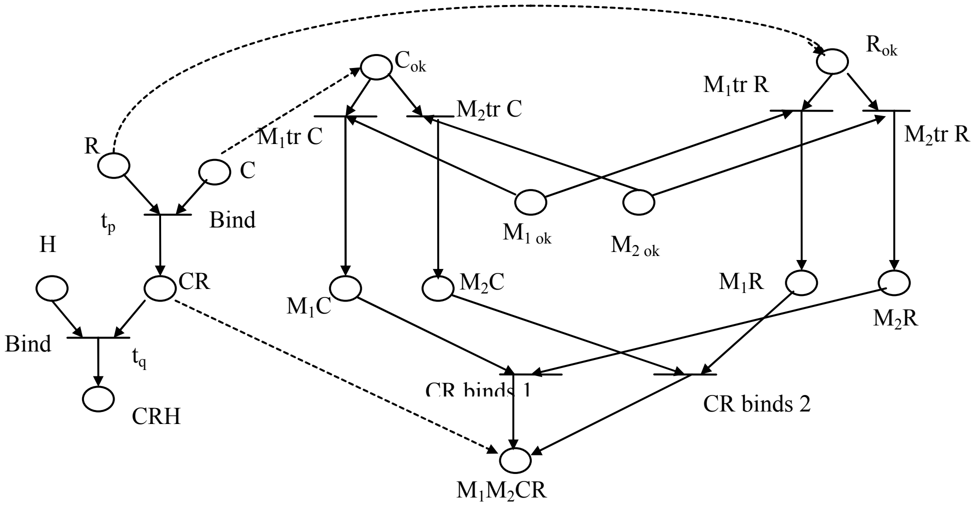

Fig.4 Refining transition t_p

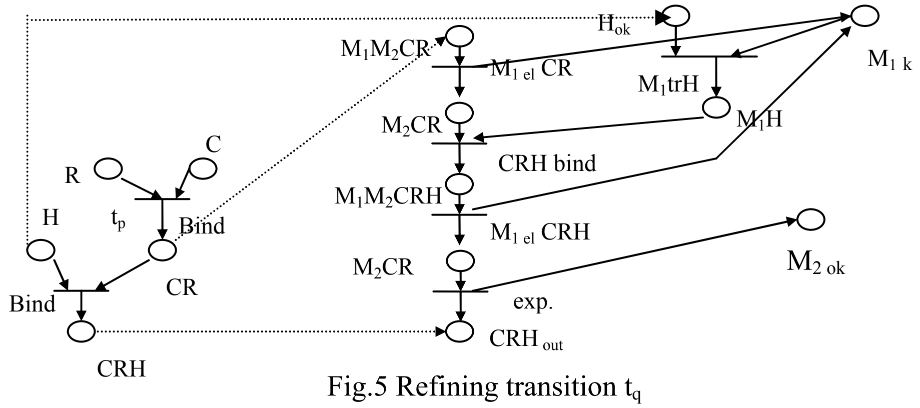

Fig.5 Refining transition t_q

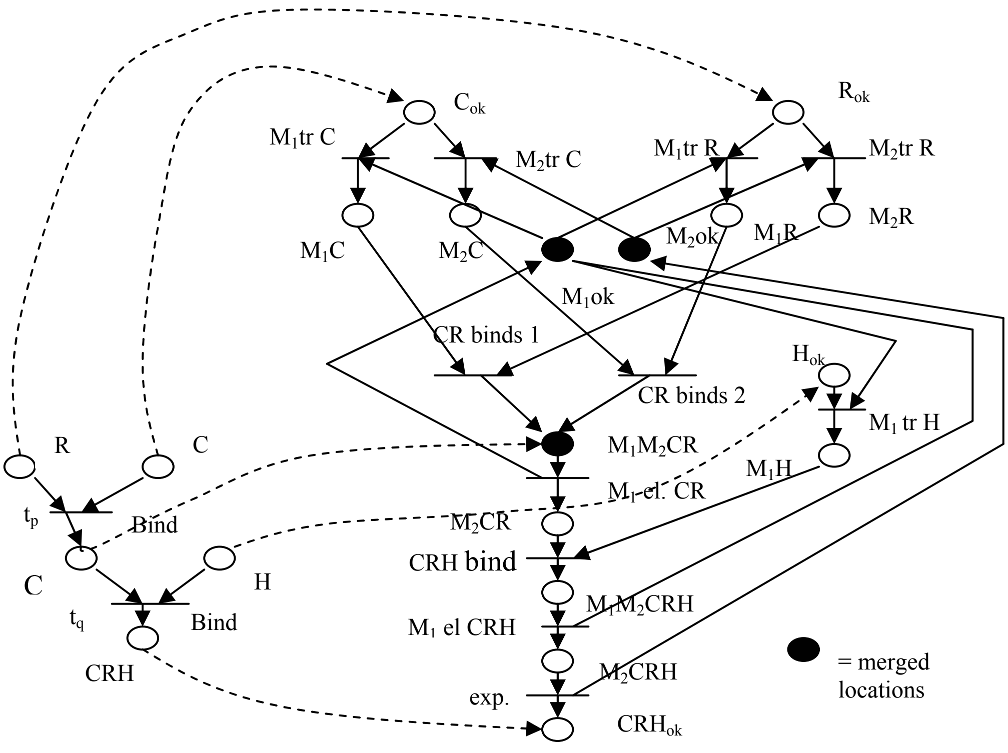

Fig.6 The G_0PN for the shunting system

In Fig.4 the refinement of the places corresponding to transition t_p of the control plan follows these expressions:

$m = \{(C, C_{OK}), (R, R_{OK}), (CR, M_1\text{-}M_2CR)\}$
$P_{IC}^{\prime} = m(I(t_p)) \cup P_{IC} = P_{IA} \cup P_{IC} = \{C_{OK}, R_{OK}, M_{1OK}, M_{2OK}\}$
$P_{OC}^{\prime} = m(O(t_p)) \cup P_{OC} = P_{OA} \cup P_{OC} = \{M_1\text{-}M_2CR\}$

Where *m* is an injective function from places in assembly plan and places in control plan, and $P^{\prime}$ are the corresponding locations for transitions $t_{p.}$ Analogous, we define the refinement of the places corresponding transition t_q of the control plan:

$m = \{(H, H_{OK}), (CR, M_1\text{-}M_2CR), (CRH, CRH_{out)}\}$
$P_{IC}^{\prime\prime} = m(I(t_q)) \cup P_{IC} = P_{IA} \cup P_{IC} = \{H_{OK}, M_1\text{-}M_2CR\}$
$P_{OC}^{\prime\prime} = m(O(t_q)) \cup P_{OC} = P_{OA} \cup P_{OC} = \{CRH_{out}, M_{1OK}, M_{2OK}\}$

The merging operations of the control plans in Fig.4, and Fig.5 results in the assembled control plan shown in Fig.6, where the significance of the indices are the same as in Fig.4, and Fig.5.

V. CONCLUSIONS

In this paper, considering that two sufficient conditions for liveness are also sufficient conditions fro the reversibility, we introduced a G_oPN to model the activities in a FMS. In order to do this, a theory that synthesizes Petri nets for modeling AMS's with shared resources has been reviewed. It merges SCSM modules through the common transitions to construct a system model with reversibility property enhanced. In order to construct a reversible G_oPN we introduced an algorithm, which can be used in a unified modeling technology. Such research increases the integrability of models with different behaviors. A G_oPN for assembly hierarchy of shunting operations in railway systems was chosen for exemplifying the given method for modeling FMS's. We shown that FMS properties in the assembly plan are mentained in the control plan, by simply following the given algorithm forreversibility controller existance in a G_oPN.

REFERENCE

[1]. A. Swaminathan, K. S. Barber (1996), *An experience based assembly sequence planner for mechanical assemblies,* IEEE Trans. On Rob. And Autom., vol. 12, no. 2, pp. 252-267
[2]. V. N. Rajan, S. Y. Naf (1996), *Minimal precedence constraints for integrated assembly and execution planning*, IEEE Trans. On Rob. And Autom., vol. 12, no. 2, pp. 175-186
[3]. M. D. Jeng, F. DiCesare (1995), *Synthesis using resource control nets for modeling shared-resource systems*, IEEE Trans. On Rob. And Autom., vol. 11, no. 3, pp. 317-327
[4]. T. Murata (1989), *Petri nets: properties, analysis and applications*, Proc. IEEE, vol 77, pp. 541-580
[5]. L. Wang, X. Xie (1996), *Modular modeling using Petri nets*, IEEE Trans. On Rob. And Autom., vol. 12, no. 5, pp. 800-809
[6]. M. D. Jeng (1997), *Petri nets for modeling automated manufacturing systems with error recovery*, IEEE Trans. On Rob. And Autom., vol. 13, no. 5, pp. 752-760
[7]. M. C. Zhon, K. Venkatesh (1999), *Modeling, simulation and control of FMS – A Petri net approach*, River Edge, NJ: World Scientific
[8]. C. Ciufudean, G. Mahalu, G. Anastasiu (2001), *Petri net representation of shunting operations*, Advances in Electrical and Computer Engineering, vol. 1(8), no. 2(16), pp. 25-28