Controlling a flexible manufacturing system using a centralized supervisor based on parallel composition of Petri networks

Mihai Popescu*, Dan Ungureanu-Anghel*, Ioan Filip* and Eugen Raduca**

*University Politehnica Timisoara, Faculty of Automation and Computer Science, Timisoara, Romania
**University “Eftimie Murgu” Resita, Electrical Engineering and Informatics Department, Resita, Romania
mihai.popescu@aut.upt.ro, dan.ungureanu@aut.upt.ro, ioan.filip@aut.upt.ro, e.raduca@uem.ro

Abstract— The design and analysis of some algorithms for the implementation of supervisors corresponding to centralized systems using Petri networks is approached in this article. Starting from the basis of supervised control, namely, monitoring and controlling of discrete event systems via a central controller (supervisor), the paper proposes a method for supervised control using Petri net type models based on parallel composition of subsystems. To illustrate and validate the proposed method, a flexible manufacturing system based on a transport system with seven nodes is used.

I. INTRODUCTION

The supervisor notion was introduced by P.J. Ramadge and W.M. Wonham in the context of supervision theory, developed in 1987 [1][2]. The notion of supervised discrete event system (DES) control was approached later in many papers [3][4][5][6]. A supervised DES control concept based on automata type models is presented in [7]. In addition to centralized supervised control, numerous papers address topics related to a supervised control with the help of supervised hierarchical control methods [8].

II. THE BASIC STRUCTURE USED IN SUPERVISED CONTROL

Controlling of complex automated systems (e.g. flexible manufacturing systems, transportation systems, etc.) requires a hierarchical (multilevel) control structure starting from the lowest hierarchical level (represented by sensors and transducers, execution elements, etc.) all the way to the software which is used to implement the complex algorithms necessary for supervised control [9][10].

For monitoring/control of various processes considered as being DES, it is necessary to use a supervisory controller (supervisor) which can be used not only to display the state of the process but also to display information provided by other components, necessary to generate the control commands [11][12][13].

The behavior of the supervisory controller depends on two types of information: conditions and events that are referred to as the controller inputs.

The state of a supervisory controller can change:
- If a condition is true - it can be expressed through logical variables corresponding to internal or external values. The external conditions relate to the state of the system, and can be characterized by predicates that are true or false propositions and can be modified by changing some logical variables.
- When undergoing an event - this refers to changing the system state (a change called event). A technical process driven by events consists of a lot of resources that are used to perform a sequence of operations. The completion of each operation requires the allocation of one or more resources that are freed up once that respective operation is finished.

The problem of supervisory control of a certain process consists in the fulfillment of the following operating conditions:
- The correct succession of operations;
- The correct allocation and release of the necessary resources needed for each operation;
- Performing a specific type of service as soon as the necessary resources for that operation are available;
- Repeatability of providing the service, without circular blockages due to the shared use of some resources.

Fig. 1 presents a general control structure which satisfies the imposed functioning conditions.

A supervisor is a unit which oversees and guides the behavior of a controlled discrete event subsystem. For the control structure represented in Fig 1, from the multitude of possible state trajectories of the system, the supervisor will select those that meet certain imposed conditions. The supervisor does not create new possible developments, it will simply choose from the existing ones. The basic role of a DES supervisor is to prevent the development of unwanted events in the supervised system by introducing certain specifications. These specifications can be of two types:

![Figure 1. The control structure](image-url)
- State avoidance problems, where the objective is to avoid some of the states;
- Sequence avoidance problems, which aim to avoid certain undesirable sequences of events.

Avoidance of certain states, or of sequences of events, is equivalent to the introduction of constraints. Consequently the supervisor must be able to introduce these constraints and can it can model an online controller.

The possible functions of the supervisor are:
- To prevent the system from entering certain forbidden states (avoid deadlock states);
- To prevent the execution of unwanted sequences;
- To force the execution of desired sequences from certain states;
- To force reaching of some desired states from given states or
- To resolve conflicts.

The supervisor must guide the subsystem towards several possible developments. To accomplish this task, there must be a choice criteria which can be implemented. There are several possible solutions for this purpose.

If the chosen solution is based on an optimal supervisor then the following guidelines must be met:
- A performance criteria must be specified;
- The performance criteria must be minimized or maximized;
- The information is transmitted towards the supervisor so it can determine the best evolution based on the most performant state trajectory of the closed loop system.

If certain decision rules have to be implemented, then a way to describe these rules must exist.

III. SUPERVISED CONTROL USING PETRI NET TYPE MODELS BASED ON PARALLEL COMPOSITION OF SUBSYSTEMS

The basic principle of the method starts from the definition of all the subsystems that make up the entire assembly and modeling them so that the resultant Petri Net type models will satisfy all the requirements in terms of structure and behavior. After defining the control specifications, on the basis of the Petri net type models parallel composition a \( PN_{Sist} \) system will result. However this is not yet the desired supervisor because it may contain blocking positions or uncontrollable transitions. After the \( PN_{Sist} \) system is obtained, the existence of bottlenecks or uncontrollable transitions is analyzed, thus moving towards the elimination of these situations.

The necessary steps of this method are presented in algorithm 1 (see in Table I. Petri net type supervisor synthesis through parallel composition).

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>All Petri net type models corresponding to the subsystems as well as their control specifications are defined</td>
</tr>
<tr>
<td>Step 2</td>
<td>All sub models are analyzed separately, so that they are viable in terms of structure and behavior</td>
</tr>
<tr>
<td>Step 3</td>
<td>All defined subsystems are parallel composed together with the control specifications</td>
</tr>
<tr>
<td>Step 4</td>
<td>The set of accessible and inaccessible (blocking generators) states is determined</td>
</tr>
</tbody>
</table>

A supervised control of a transport system with accumulation area is considered in Fig. 2.

![Figure 2. The structure of the transport system](image)

Maximum “jam” capacity as well as the initial state of the “jams” is presented in Table II (the notion of “jam” refers to an accumulation area). As Table II suggests, there are 35 carriages in the system and the maximum space available is for 47 carriages. As a result of this load the system is functional, and there is no risk of operational blocking situations.

By analyzing the structure of the transport system considered in Fig. 2 it can be noted that the system is composed of six nodes, of which two type 1 (N1 and N7), three type 4 (N2, N4 and N6), one type 3 (N3) and one type 2 (N5).

<table>
<thead>
<tr>
<th>Jam ID</th>
<th>Max capacity</th>
<th>Initial capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jam 1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Jam 2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam 3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam 4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam 5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Jam 6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam 7</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam 8</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Jam 9</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jam10</td>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

![Figure 3. The Petri net structure corresponding to the proposed system](image)
The Petri net structure corresponding to the modeling of the proposed system with the use of sub models is presented in Fig. 3.

The system’s operation is based on the following specifications:
1) The N2 node (type 4 node) will operate so that in case the result of the scan is different than the prescribed value (for example 2) all corresponding carriages will be moved to “jam” 3.
2) The N3 node (type 3 node) will operate on the principle of rotation. A carriage is released in sequence from each of the stoppers. If a “jam” does not contain any carriages then the next “jam” is analyzed.
3) The N4 node (type 4 node) will operate so that in case the result of the scan is different than the prescribed value (for example 6) all corresponding carriages will be moved to “jam” 7.
4) The N5 (type 2 node) operates on the same principle as node 3.
5) The N6 node (type 4 node) will operate so that in case the result of the scan is different than the prescribed value (for example 9) all corresponding carriages will be moved to “jam” 10.

Fig. 4 presents the Petri net type models corresponding to the imposed specifications for type 4 nodes (N2, N4 and N6).

Starting from the construction mode of Petri net PN_Sist the analysis from blocking point of view is done in steps, actually by using sub models. This analysis is possible because the sub models that are used are interconnected between them so that their properties are not influenced by each other. Base sub models are analyzed in [14][15][16].

The basic structure for the nodes was modified by adding the agreed specifications and for that reason a structural and behavioral analysis of the nodes is required. The structure of nodes N2, N4 and N6 is identical therefore the analysis was performed only on node N2. The modified structure corresponding to N2 node (type 4 node) is presented in Fig. 7.
In Fig. 7, the transition $t_5$ acts as a generator therefore supplying the considered Petri net with tokens. Because for this network only the possibility of blockage is relevant, the following network presented in Fig. 8 is analyzed only based on the coverability tree.

![Figure 8. The coverability tree of the modified Petri net for the type 4 node](image)

Where (Fig. 8):

- $M_0 = [1,0,0,1,1,0]$; $M_1 = [0,1,0,0,1,0]$; $M_2 = [0,0,1,0,0,1]$;
- $M_3 = [2,0,0,1,1,0]$; $M_4 = [0,0,0,1,1,0]$; $M_5 = [1,1,0,0,1,0]$;
- $M_6 = [1,0,1,0,0,1]$; $M_7 = [2,1,0,0,1,0]$; $M_8 = [2,0,1,0,0,1]$.

The coverability tree structure shows that the considered Petri net, the subnet corresponding to the N2 node, is not in deadlock. This result is valid for nodes N4 and N6 because they are type 4 nodes as well with the same functional specifications imposed on the system.

The N3 node (type 3 node) is analyzed next in order to verify the existence of deadlocks as a result of the introduction of the control specifications. The structure for the considered subnet is presented in Fig. 9.

![Figure 9. The modified structure of the N3 node – type 3 node](image)

The $t_8$, $t_9$ and $t_{10}$ nodes are generator transitions and they supply tokens to the network. The N5 (type 2 node) node is analyzed next in order to confirm the existence or inexistence of deadlocks as a result of the introduction of the control specifications. The considered subnet structure is presented in Fig. 10.

![Figure 10. The modified structure of the N5 node – type 2 node](image)

Because the considered sub models do not create deadlocks in the overall operation of the system that resulted from parallel composition, it can be concluded that the Petri network $PN\_Sist$ (Fig. 6) represents the desired supervisor model.

IV. CONCLUSION

This article proposes and validates a method which can be used for the control and supervised observation of a discrete event system by using parallel composition of Petri type networks. The starting point of the proposed strategy is to design a centralized supervisor with the task of controlling and monitoring a DES. The chosen Petri type net is suitable to control a flexible manufacturing line which uses a transport system that consists of seven nodes. A multilevel control structure is used for this purpose which starts from the lower level (sensors and actuators) and it reaches the top layer represented by complex software control algorithms.

The problems which can occur during the supervised control of a system, the specifications and the possible functions that the supervisor can perform during the operation of the system are presented by the authors. A performance criterion was used in order to evaluate and validate the accuracy of the supervisor. The use of the parallel composition method implies using Petri net models which satisfy the structural and behavioral requirements of the studied system. Later on, the control specifications are defined and through parallel composition the Petri net type models, a resulting system called $PN\_Sist$ is formed. This resulting system represents a draft of the desired supervisor. The next step is the elimination of deadlocks or uncontrolled transitions by using the Petri net type supervisor synthesis algorithm through parallel composition. It can be noted that after the synthesis algorithm was applied all the deadlocks were eliminated, therefore it can be concluded that the Petri net $PN\_Sist$ which resulted from parallel composition of the considered sub models represents the desired supervisor, thus validating the proposed strategy.
REFERENCES


