Management and Production Engineering Review
Volume 7 • Number 2 • June 2016 • pp. 62–72
DOI: 10.1515/mper-2016-0018

SYNTHESIS OF PETRI NET BASED MODEL OF A DISCRETE EVENT MANUFACTURING SYSTEM FOR NONLINEAR PROCESS PLAN

Adam Słota[1], Jerzy Zając[1], Marimuthu Uthayakumar[2]

[1] *Cracow University of Technology, Production Engineering Institute, Poland*
[2] *Kalasalingam University, Department of Mechanical Engineering, India*

Corresponding author:
Adam Słota
Cracow University of Technology
Production Engineering Institute
Al. Jana Pawła II 37, 31-864 Kraków, Poland
phone: (+48) 12 374-32-50
e-mail: slota@mech.pk.edu.pl

Received: 2 March 2016
Accepted: 16 May 2016

Abstract
This work presents a modelling approach for nonlinear process plan (NLPP) implementation in discrete event manufacturing system (DEMS). NLPP is used for the building of the modular structure of an Object Observable Petri Net model of DEMS. The general capabilities of DEMS are defined by resources' operation templates and the transition incidence relation. Based on system specification and NLPP executed in the system, the modular model of DEMS is defined. The required steps for constructing a modular model through the integration of resource models are presented. The proposed approach to modular modelling is illustrated by means of a sample DEMS and an example of NLPP.

Keywords
manufacturing system, Petri net, nonlinear process plan, modular modelling.

Introduction

Competitiveness of contemporary markets puts pressure on working out still more effective manufacturing methods and production management. The production effectiveness can by improved directly by cost reduction and shortening production cycle time. Two important stages of the production cycle, which greatly influence the production effectiveness, are process planning and scheduling.

Process plan defines methods and resources required to manufacture a product. It divides the process into a set of operations, defines feasible sequence of operations and determines all the information concerning raw materials, machines, tools, fixtures, machining parameters. Though in industrial practice mostly manual process planning method is used, much research is devoted to computer aided process planning systems (CAPP) [1, 2]. CAPP systems utilizes two main approaches: variant and generative. Nonlinear process plans (NLPP), also referred to as alternative process plans, define different ways of a product manufacturing. They take into account possibility of using different machines, tools or operation structures. Availability of process plan alternatives enables production cost estimation and making decisions to increase production effectiveness [3].

Scheduling assigns operations defined in process plan to specific resources and determines operation execution time. It is commonly acknowledged, that scheduling a set of jobs (each consisting of a set of operations, each of which requires uninterrupted processing on a given machine) on a set of machines is NP hard problem. The scheduling complexity grows when other resources, like tools, fixtures, operators are to be taken into account [4]. The problem to solve is even more difficult when NLPP is considered. In the presented research, to obtain semi-optimal solutions, heuristic approaches are used [5, 6]. Addition-

ally only machines, as processing resources, are considered in the scheduling process, while all the equipment of material handling subsystem (AGV, AS/RS, robots) are neglected [7, 8].

In a traditional approach process planning and scheduling were two tasks executed in a sequence: the schedule was built for a ready process plan. To increase effectiveness of both process planning and scheduling the need of integration of these two tasks was recognized and many researchers approached the problem. References [9–11] present extensive reviews of the research in the area of process planning and scheduling integration. Authors classify all the approaches into three main groups: nonlinear, closed-loop and distributed process planning. In nonlinear approach alternative process plan is generated offline assuming static shop floor situation and then, based on a defined criteria, one of the alternatives is chosen for execution. Closed-loop methodology relies on dynamic feedback from production scheduling to process plan generation. Process plans generated with the use of feedback information, which include data about the current state of the manufacturing resources, are feasible in terms of the availability of the production facilities. Distributed approach means concurrent process planning and scheduling, executed in two phases: preplanning and final planning. The methods mentioned above aim at generation reliable and effective production scheduling. However in real manufacturing systems there are unpredictable situation, referred to as disturbances, such as machine breakdown, tool damage, shortage of power supply [9]. This means that some schedules, worked out as optimal, at the time when manufacturing starts, have to be modified. Survey shows that 20–30% of schedules has to be modified, what means jobs redirection to alternative machines, to meet the desired objective [12]. The conclusion is that effective production management cannot be solved even by highly integrated process planning and scheduling. An online, real time mechanism of control decision making is required to cope with situations which cannot be foreseen at the time of process planning and scheduling. A solution may be a simulation based decision making mechanism, which simulates system behaviour in a defined time window, taking into account local conditions. Then, based on the simulation result locally optimal decisions are made [8, 13–15].

Variation and changes in products along with the application of NLPP add more complexity to production processes. Dealing with such complexity requires models that are capable of representing a very diverse set of situations in discrete event manufacturing systems (DEMS).

To effectively build and manage real scale DEMS models a modular approach has been developed [9, 16]. The model in this case is considered as a collection of modules that, in using different symbols and mathematical relations, represents the knowledge and data of the manufacturing system. Basic advantages of modular models are [17]: genericity (module definition enables module instantiation with different parameters), modularity (complex systems are represented as a set of smaller modules, managed independently, but integrated together to function as a whole), reusability (a module may be used to model a component with a similar behaviour after only small modification) and abstraction (to analyse whole system behaviour modules may be seen as black boxes with only interfaces to represent modules' interactions). Modular models are suitable for agent based approaches to DEMS management [4].

For modelling DEMS different languages have been used, like Unified Modelling Language (UML) [18, 19] or automata [20], but one of the most recognized and widely used is Petri net (PN) formalism [21–26]. The PN language is so popular because it provides possibility of modelling parallel and alternative processes as well as sequential precedence constraints. The model is not only a static structure but it also covers state evolution in time via marking distribution changes. Additionally well defined methods of structural property analysis, like: boundness, liveness, place and transition invariants are available. And last but not least PN provides visual representation of the model, what substantially simplifies model creation and analysis. PN appeared also to be an effective tool for building modular models. Reference [27] outlines the issues of modular state space and compositional verification. Reference [16] uses PN to model behaviour of the system components. For each module input/output conditions and events are defined and modules are connected by condition and event arcs. Parallel machines, transfer chains, assembly and disassembly module models based on PN are presented in [28]. High level extension of PN – fuzzy coloured PN with stochastic time delay is introduced in [29] for building parametric modules.

PN are also used in the area of modelling NLPP. Precedence relations and alternative process branches are naturally represented in PN model. Incidence matrix representation of PN enables straightforward usage of integer linear programming for process plan analysis [6]. A subclass of PN named process planning net (PP-net) which is an ordinary acyclic safe PN with some limitations defined for PN structure and marking is introduced in [30].

From the presented above research analysis it can be concluded that: (i) integration of NLPP and scheduling is indispensable for flexible manufacturing implementation; (ii) due to dynamic nature of DEMS and unpredictable, at the time of scheduling, situations a real time working control system is required for online job rerouting to meet the defined production objectives; (iii) control system of DEMS requires a mathematical model which integrates DEMS structure description (system components and operation rules) and information about all the processes (defined by NLPP), executed in DEMS; (iv) to get manageable models of real scale DEMS modular approach is used effectively; (v) PN is a suitable formalism to model both DEMS operation rules as well as NLPP. Though much research is devoted to the subject, the authors notice the gap which concerns definition of a model which comprises data defining both NLLP and DEMS specification and is suitable for online control and scheduling.

The contribution of the paper is the algorithm of creation DEMS modular model containing information necessary for dynamic job rerouting in case of disturbances. The model is generated on the basis of templates of DEMS components operation and their interactions in terms of material flow. Information defining NLPP, obtained from a CAPP system, are also included in the model. For all the modelling stages (DEMS components operation templates, NLPP and resulting DEMS modular model) an extension of PN – object observable Petri net (OPN) is used. The model is designed to be used by a software for the simulation and control of DEMS. Modelling issues considered in the paper are limited to machining systems.

The remainder of this paper is organized as follows. Next section presents PN formalism. Then modular model building procedure is presented – it comprises: DEMS specification, a PN model of NLPP and algorithm of modular model creation. The paper is summarized with an example of modular model building.

Petri net formalism

PN is a language developed for the modelling and analysis of dynamic discrete event systems. *PN* is defined as a directed bipartite graph with a set of places P and a set of transitions T as nodes. The flow relation F (directed arcs) defines the relation between places and transitions. The graph is supplemented with additional data by means of weight function W and initial marking function M_0. Places represent conditions, transitions represent actions and places' marking (tokens) represent condition fulfilment. A transition may be fired when all the conditions defined by its input places are satisfied (a place p is an input place of a transition t if $(p,t) \in F$). Transition firing leads to a change in the condition fulfilment represented by input and output places of the fired transition (a place p is an output place of a transition t if $(t,p) \in F$). This general interpretation makes *PN* suitable for modelling and analysis of a wide range of discrete event systems: from computer systems to biological systems.

In the case of DEMS modelling a precise interpretation to *PN* model elements may be assigned and model building rules may be defined. DEMS consists of a set of NR resources (machine tools, robots, AGVs, storage systems, etc.) $R = \{R^1, R^2, \cdots, R^{NR}\}$. Parts of NI types $PT = \{PT^1, PT^2, \cdots, PT^{NI}\}$ are produced in the system. All the processes executed in the system may be divided into a set of activities, which comprise parts' processing operations and parts' flow operations. These activities are executed by system resources. When resources start the execution of an activity they become unavailable for starting any other activity until the current activity is completed.

To reflect directly the structure of DEMS in its *PN* model, and thus enable modular modelling, a new class of *PN*, named Object Observable Petri Net (*OPN*), is introduced. The following interpretation of *OPN* elements is assumed:

- places represent system resources' and parts' states, which are defined as the availability to start an activity,
- one place can represent a state of one resource or a state of a set of parts of the same type,
- transitions represent activities executed by the system's resources,
- tokens represent physical elements of the modelled system – resources and parts,
- weight function defines the number of resources or parts required to start an activity and the number of resources or parts released when the activity finishes.

Taking into account these interpretation rules *OPN* is defined as a seven-tuple $OPN(P, T, F, W, M_0, J, \tau)$ [30]. The first four elements: P – a set of places, T – a set of transitions, F – flow relation, W – weight function are defined as in classical *PN*. M_0 is the initial marking function defined for places and transitions (1)

$$M_0 : P \cup T \to N \quad \text{and} \quad \forall t \in T \;\; M_0(t) = 0. \qquad (1)$$

J is the objects' numeration function. It assigns a natural number to each place (2) and splits the set P into disjoint subsets P_j (3). Each subset P_j represents states of one resource or states of parts of the same type. Thus, each place is identified by two numbers: for example $p_{k,l}$ denotes a place number k in a subset of places P_l

$$J : P \rightarrow N, \tag{2}$$

$$P = \bigcup_j P_j \quad \text{and} \quad \bigcap_j P_j = \emptyset. \tag{3}$$

If a place $p_{a,j} \in P_j$ is an input place of a transition t ($p_{a,j} \in {}^*t$), there is exactly one place $p_{b,j} \in P_j$, which is an output place of the transition t ($p_{b,j} \in t^*$) (4)

$$\forall t \in T \forall P_j \, |{}^*t \cap P_j| = |t^* \cap P_j| \leq 1. \tag{4}$$

If there are redundant resources in the system and the same route of different parts in the system an extension of OPN – Colored OPN is introduced [31]. In such a case subset P_j represents states of a set of redundant resources or states of a set of parts of different types.

Weight function values defined for arcs which connect places $p_{a,j}$ and $p_{b,j}$ with transition t are the same (5)

$$\begin{gathered} \forall (p_{a,j}, t) \in F \wedge (t, p_{b,j}) \in F, \\ W\left(t, p_{b,j}\right) = W\left(p_{a,j}, t\right), \end{gathered} \tag{5}$$

τ defines the firing time for transitions (6)

$$\tau : T \rightarrow R. \tag{6}$$

The transition state (enabled, not enabled) is calculated according to the rules commonly used in PN. Firing of an enabled transition t in OPN is executed in three stages according to the rules defined for transition timed PN:

- tokens in quantities defined by a weight function are removed from input places and are attached to the transition t,
- net marking remains unchanged for the time $\tau(t)$,
- tokens in quantities defined by a weight function are detached from transition t and moved to its output places.

OPN definition and transition firing rules guarantee that the firing of any transition does not change the number of tokens in the model. This is consistent with the assumed interpretation of OPN elements. Direct representation of DEMS components (resources and parts) makes OPN suitable for modular modelling [32].

Modular structure of an OPN based model of DEMS

System specification

The typical modelling process of DEMS can be described using a two-stage procedure. At the first stage system members are specified by dividing the manufacturing system into modules with limited buffer capacity. The most natural granularity level is the level of manufacturing equipment like machine tools, robots, AGVs, storage systems etc., that have individual control units. This particularly relates to models for shop floor control. Moreover, at this stage, elementary activities (operations) performed by the system members are also identified. Some of the activities are parts' processing operations while the other are parts' flow operations. Parts' processing operations are operations that are necessary to transform raw materials into final products. These operations are defined during process planning. Parts' flow operations are responsible for the transportation of parts during the execution of the production process. A part flowing through the system requires a single unit of buffer capacity at each module necessary to perform a feasible sequence of operations (route). The second stage of the modelling process concerns the determination of a system's operation rules that ensure the proper operation of the system by checking whether the capacity for each module has not been exceeded. This protects the system against collision like e.g., loading a machine tool that is already loaded. At this stage essential information is the parts' process plans executed in the system which defines the sets of operations with precedence constraints and possible alternatives. The integration of DEMS specification with process plans information results in a model which may be used for offline DEMS operation analysis as well as for online control. The result of this stage is an DEMS model developed by a synthesis of modules' models.

All the DEMS resources defined by the R set may be divided into two disjoint subsets: RP – processing resources (machine tools) and RF – part flow subsystem resources (robots, AGVs, storage systems). It is assumed that at any stage of the process execution processed parts are associated with at least one resource. In the initial state S_{start} parts are associated with a resource $R^s \in RF$, in the final state S_{end} parts are associated with a resource $R^e \in RF$. Permissible parts flow is defined by a resource incidence relation $RI \subset R \times R$. A pair $(R^i, R^j) \in RI$ means that resources R^i and R^j are capable (taking into account their kinematic structure, workspaces' location, tooling and manipulation equipment, con-

trol software, etc.) to transfer parts directly from R^i to R^j without the use of other resources.

For each resource an operation template is defined. The template describes, with the use of OPN language, all the possible sequences of activities for processing a virtual part. The resource operation template $ROPN^j(P_R^j, T_R^j, F_R^j, W_R^j, M_{0R}^j, J_R^j, \tau_R^j)$ for resource R^j contains the following elements:

- a set of input transitions T_{RI}^j representing parts' transfer at the side of a receiving resource; for each R^k resource such as $(R^k, R^j) \in RI$ an input transition t_{RIk}^j is created,
- a process transition t_{RP}^j representing a part processing operation (only for templates of resource $R^j \in RP$),
- a set of output transitions T_{RO}^j representing parts' transfer at the side of transferring resource; for each R^l resource such as $(R^j, R^l) \in RI$ an output transition t_{ROl}^j is created,
- a set of auxiliary transitions T_{RA}^j which represent activities that are not directly connected with part processing or inter resource part transfer, these transitions represent activities of resource maintenance or reconfiguration for processing parts of different types,
- a place p_{RId}^j representing an idle state from which the resource can start processing a part of any type (in general a change in the types of processed parts may require a resource tooling reconfiguration),
- a set of places P_R^j which represent resource's states during an operation,
- a flow relation which connects the places and transitions and weight function defined according to the interpretation assumed for OPN.

A sample resource operation template for R^j resource is presented in Fig. 1. In the template auxiliary transitions and places associated with them as well as arcs are marked by dashed lines.

An operation template of an R^j resource has to contain one p_{RId}^j place and at least one of the transition sets T_{RI}^j or T_{RO}^j has to be a nonempty set.

A template for an R^j resource is integrated with the templates of other R^k and R^l resources (such as $(R^k, R^j) \in RI$ or $(R^j, R^l) \in RI$) by the transition incidence relation (7)

$$TI^j \subset (T_{RO}^k \times T_{RI}^j) \cup (T_{RO}^j \times T^l{}_{RI}). \tag{7}$$

The relation defines the pairs of transitions which represent the same transfer activity. For input and output transitions of a given template the relation is unique: for every input transition t_{RIk}^j in template for R^j resource there exists exactly one resource R^k with one output transition t_{ROj}^k in its template which matches t_{RIk}^j and for every output transition t_{ROl}^j in template for an R^j resource there exists exactly one resource R^l with one input transition t_{RIj}^l in its template which matches t_{ROl}^j.

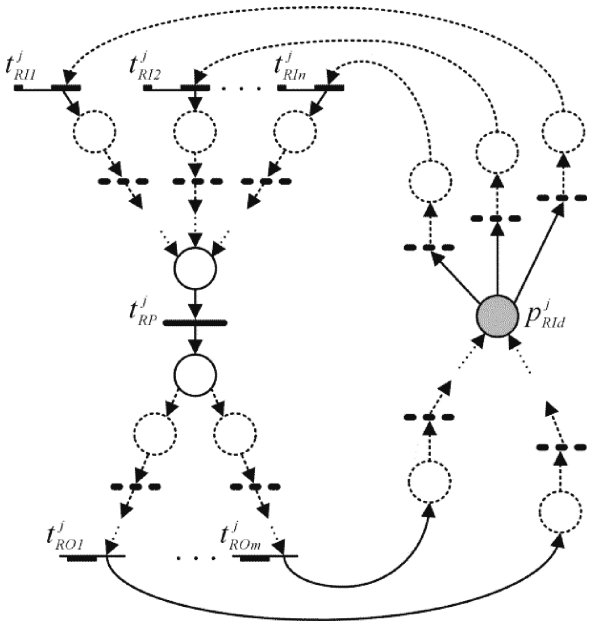

Fig. 1. A graphical view of a sample resource operation template.

NLPP specification

Types of parts produced in DEMS are defined by the PT set. For parts of each type $PT^i \in PT$ a NLPP is defined. The process plan for PT^i defines a set of NW operation alternatives $OA^i = \{OA_1^i, OA_2^i, \ldots, OA_{NW}^i\}$ with precedence constraints. Issues concerning generation of alternative operations for machining process plans are discussed in [33]. For each operation the processing resource is assigned by the operation-resource relation $OR^i \subset OA^i \times RP$. A pair $(OA_j^i, R^l) \in OR^i$ means that operation OA_j^i for parts of PT^i needs to be processed on the resource R^l in an interrupted period of time without the need to use any other resource. Precedence constraints allow alternative sequences of operations which transform a part from an initial state S_{start} to a final state S_{end}.

A NLPP for PT^i parts may be defined using OPN language as a process template $POPN^i(P_P^i, T_P^i, F_P^i, W_P^i, M_{0P}^i, J_P^i, \tau_P^i)$. The process template is a non-cyclic net. Different branches of the net define alternatives of process execution. A set of NW transitions T_{PO}^i represents the set of operations OA^i. It is assumed that the transition t_{POj}^i in process template represents the operation alternative OA_j^i which is executed by the resource $R^j \in RP$. A set of NF places P_P^i, represents a part's states. For each

transition t^i_{POj} there is only one input place and one output place (8)

$$\forall t^i_{POj} \in T^i_{PO} \left|{}^*t^i_{POj}\right| = \left|t^i_{POj}{}^*\right| = 1. \qquad (8)$$

For place p^i_{Pj}, representing part's intermediate states which allow alternative branching of the process, the number of output transitions equals the number of alternative operations possible to start from this part state. For place p^i_{Pj}, representing part's intermediate states which are results of more than one operations, the number of input transitions equals the number of alternative operations which produce part in this state. There is one place p^i_{PS0}, which represents initial part state S_{start}, with no input transitions and one place p^i_{PSNF}, which represents final part state S_{end} with no output transitions.

Models of DEMS resources for NLPP

The proposed DEMS model which comprises a description of both system resources and NLPP has a modular structure. A resource operation template defines all the possible cycles of activities which can be executed by a given resource of DEMS. The model of a particular resource is an instance of its operation template customized to execute a defined set of process plans. Some cycles of the template may be omitted in the model, as not used for the execution of defined process plans, while some may have multiple copies required to execute different alternatives of defined process plans. The DEMS model, based on the OPN defined for the PT^i parts is constructed in the following steps:

(A) Checking the feasibility of process execution taking into account processing resources. For each transition t^i_{POj}, representing OA^i_j operation identify resource $R^j \in RP$ such as $(OA^i_j, R^j) \in OR^i$.

(B) Finding out the part flow paths between machines which execute two successive operations in a process plan. For each pair of transitions $\left(t^i_{POk}, t^i_{POl}\right)$ in the process template $POPN^i$ such as $\left(t^i_{POk}, p^i_{Pl}\right) \in F^i_P$ and $\left(p^i_{Pl}, t^i_{POl}\right) \in F^i_P$ identify a subset RI^i_{kl} of resource incidence relation RI that define all the possible paths for parts from R^k to R^l such as $RI^i_{kl} \subset (RF \cup \{R^k\}) \times (RF \cup \{R^l\})$.

(C) Finding out the part flow paths between the resource R^s, to which parts in the initial state S_{start} are attached, and machines which execute first operations in different alternatives of the process plan. For all transitions t^i_{POj} such as $(p^i_{PS0}, t^i_{POj}) \in F^i_P$ identify a subset RI^i_{sj} of the resource incidence relation RI that define all the possible paths for parts in the initial state being attached to the R^s resource to the R^j resource such as $RI^i_{sj} \subset RF \times (RF \cup \{R^j\})$.

(D) Finding out the part flow paths between machines which execute final operations in different alternatives of the process plan and resource R^e, to which parts in the final state S_{end} are attached. For all transitions t^i_{POj} such as $(t^i_{POj}, p^i_{PSNF}) \in F^i_P$ identify a subset RI^i_{je} of the resource incidence relation RI that define all the possible paths for parts from R^j to R^e with attached parts in the final state such as $RI^i_{je} \subset (RF \cup \{R^j\}) \times (RF)$.

(E) Creation models of resources responsible for parts' flow between the initial resource R^s, machines executing machining operations and final resource R^e. Resource models consist of copies of some cycles from the resource operation template. For occurrences of resources $R^v \in RF$ in different RI^i_{kl}, RI^i_{sj}, RI^i_{je} relations and each R^u and R^w such as $(R^u, R^v) \in RI^i_{kl} \cup RI^i_{sj} \cup RI^i_{je}$ and $(R^v, R^w) \in RI^i_{kl} \cup RI^i_{sj} \cup RI^i_{je}$ create in the R^v model copies of cycles from the $ROPN^v$ template which start from p^v_{RId} and contain the input transition t^v_{RIu} (for each R^u) and the output transition t^v_{ROw} (for each R^w).

(F) Creation models for machines executing operations defined in NLPP. For resources $R^j \in RP$ create in the R^j model copies of cycles from the $ROPN^j$ template for each operation alternative OA^i_j such as $(OA^i_j, R^j) \in OR^i$, a cycle should start from p^j_{RId} and contain input transitions t^j_{RIu} for all R^u resources such as $(R^u, R^j) \in RI^i_{kj} \cup RI^i_{sj}$ and output transitions t^j_{ROw} for all R^w resources such as $(R^j, R^w) \in RI^i_{jl} \cup RI^i_{je}$.

(G) Creation a model of the initial resource R^s, to which parts in the initial state S_{start} are attached. Create a model of the R^s resource by making copies of the $ROPN^s$ template which start from p^s_{RId} and contain an output transition t^s_{ROw} for each R^w such as $(R^s, R^w) \in \bigcup_j RI^i_{sj}$.

(H) Creation a model of the final resource R^e, to which parts in the final state S_{end} are attached. Create a model of R^e by making copies of the $ROPN^e$ template which start from p^e_{RId} and contain an input transition t^e_{RIu} for each R^u such as $(R^u, R^e) \in \bigcup_j RI^i_{je}$.

(I) Definition in the models of part flow resources and the processing resources states of produced parts. Each resource model created in steps (5) and (6) is supplemented with part flow information by adding between the input and output transitions places (and transitions if required) and relations to represent the state changes for the parts attached to the resource.

(J) Definition in the model of the initial resource states of produced parts in the initial state S_{start}. The supplement R^s resource model with places (and

transitions if required) and relations to represent state changes of parts between their initial state and output transitions.

(K) Definition in the model of the final resource states of produced parts in the final state S_{end}. The supplement R^e resource model with places (and transitions if required) and relations to represent state changes of parts between input transitions and parts' final state.

(L) Integration of the models of all the resources using a transition incidence relation defined in resource operation templates. In each resource model, for each copy of input and output transition, define a copy of a transition incidence relation which joins in pairs equivalent copies of input and output transitions.

Example

The procedure of building a modular model of DEMS for a defined NLPP will be presented for the sample automated manufacturing system (Fig. 2). The system consists of a storage S, an AGV, two machining stations MS1 and MS2 with input buffer storages BI1, BI2 and output buffer storages BO1, BO2 respectively. The AGV transports a set of eight, sixteen or twenty four parts on a pallet and is equipped with a manipulation device which enables pallet transfer to and from storage and buffer storages. Machining stations contain a manipulator for the loading and unloading of single parts. Dashed arrows in the diagram show the possible part flow in the system.

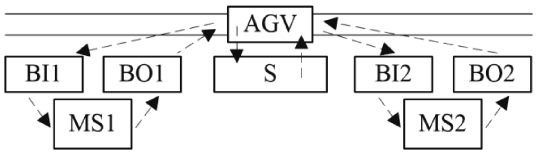

Fig. 2. A diagram of a sample manufacturing system.

The sample manufacturing system is defined by: the set of processing resources $RP = \{MS1, MS2\}$, the set of part flow resources $RF = \{S, AGV, BI1, BI2, BO1, BO2\}$, resources for parts in the initial and final state $R^s = R^e = S$ and the resource incidence relation (9)

$$RI = \left\{ \begin{array}{c} (S, AGV), (AGV, BI1), \\ (AGV, BI2), (BI1, MS1), \\ (MS1, BO1), (BI2, MS2), \\ (MS2, BO2), (BO1, AGV), \\ (BO2, AGV), (AGV, S) \end{array} \right\}. \quad (9)$$

For each resource of the sample manufacturing system a resource operation template is defined (Fig. 3). Templates of different resources are surrounded by dashed line envelopes, idle states are denoted by grey circles. To identity a transition in a resource template the following notation is used: $t^{Id}_{RType(Arg)}$. Id is the identifier of the resource tem-

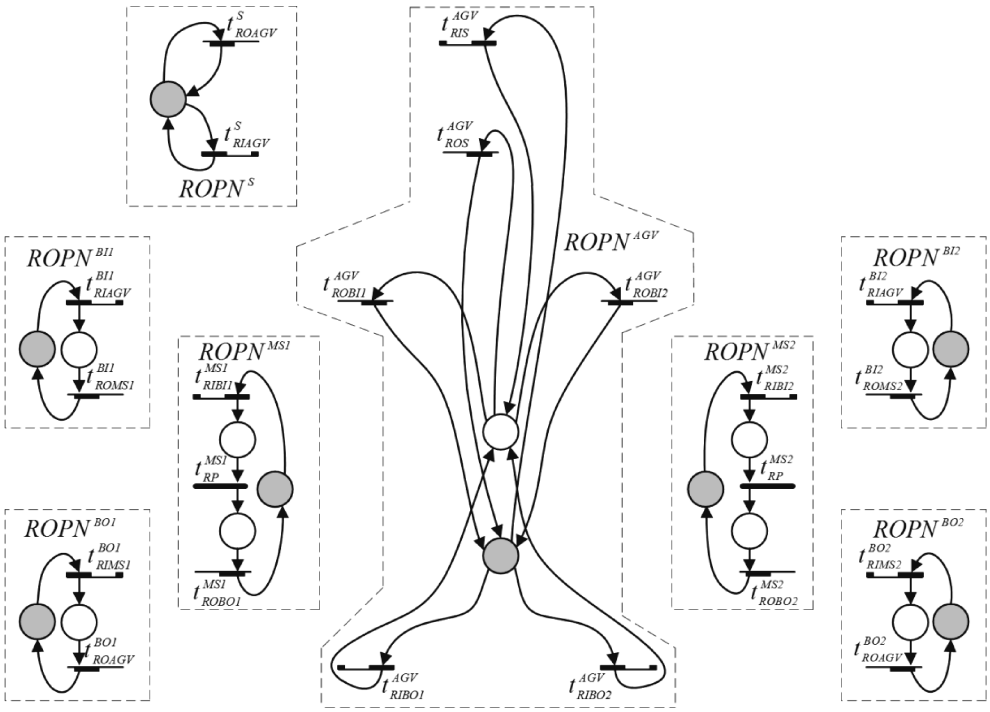

Fig. 3. A graphical view of the resource operation templates for the sample system.

plate the transition belongs to, *Type* defines transitions' type (I for input, O for output and P for process transition is used), *Arg* (used for input and output transitions) identifies the resource template which contains a corresponding counterpart of the defined transition.

A modular model of the system is constructed for a sample NLPP for a shaft of the PT^1. It is assumed that rough machining may be executed with the use of two different sets of tooling and thus two alternative operations for rough machining (OA_1^1 and OA_3^1) are defined. Alternative operations OA_1^1 and OA_3^1 produce parts in two different intermediate states. Similarly two alternative operations for finishing machining (OA_2^1 and OA_4^1) are defined. A graphical view of the sample process template $POPN^1$ is presented in (Fig. 4). In the model, the transition t_{POj}^1 represents the operation OA_j^1.

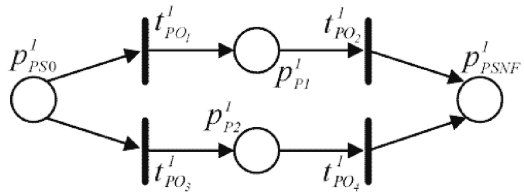

Fig. 4. A graphical view of the sample process plan.

Operation-resource relation is defined as:

$$OR^1 = \left\{ \begin{matrix} (OA_1^1, MS1), (OA_2^1, MS2), \\ (OA_3^1, MS1), (OA_4^1, MS2) \end{matrix} \right\}. \quad (10)$$

For system resources and the process specification a model is constructed according to the steps (A) to (L). For parts' flow between machines MS1 and MS2 executing pairs of two successive operations (OA_1^1, OA_2^1) and (OA_3^1, OA_4^1), the resource incidence relation subsets RI_{12}^1 and RI_{34}^1 are identified

$$RI_{12}^1 = \left\{ \begin{matrix} (MS1, BO1), (BO1, AGV), \\ (AGV, BI2), (AGV, S), \\ (S, AGV), (BI2, MS2) \end{matrix} \right\}, \quad (11)$$

$$RI_{34}^1 = \left\{ \begin{matrix} (MS1, BO1), (BO1, AGV), \\ (AGV, BI2), (AGV, S), \\ (S, AGV), (BI2, MS2) \end{matrix} \right\}. \quad (12)$$

For the flow of parts between storage S, to which parts in the initial state are attached, and machine MS1, which executes the first operations OA_1^1 and OA_3^1 in two alternatives of the process plan, the resource incidence relation subsets RI_{s1}^1 and RI_{s3}^1 are identified. Since one initial part state is considered and both operations OA_1^1 and OA_3^1 are executed by the same machine MS1 these relations are identical

$$RI_{s1}^1 = RI_{s3}^1 = \left\{ \begin{matrix} (S, AGV), (AGV, BI1), \\ (BI1, MS1) \end{matrix} \right\}. \quad (13)$$

For the flow of parts between machine MS2, which executes the last operations OA_2^1 and OA_4^1 in two alternatives of the process plan, and storage S, to which parts in final state are attached, the resource incidence relation subsets RI_{2e}^1 and RI_{4e}^1 are identified. Since both operations OA_2^1 and OA_4^1 are executed by the same machine MS2 these relations are identical

$$RI_{2e}^1 = RI_{4e}^1 = \left\{ \begin{matrix} (MS2, BO2), (BO2, AGV), \\ (AGV, S) \end{matrix} \right\}. \quad (14)$$

Then models of part flow resources (apart from storage S) are built. Let's take as an example AGV resource. On the basis of relation (11) a copy from $ROPN^{AGV}$ containing the input transitions $t_{RIS}^{AGV(I)}$, $t_{RIBO1}^{AGV(I)}$ and the output transitions $t_{ROS}^{AGV(I)}$, $t_{ROBI2}^{AGV(I)}$ is created. For relation (12) a copy of the first cycle is created with the transitions $t_{RIS}^{AGV(II)}$, $t_{RIBO1}^{AGV(II)}$, $t_{ROS}^{AGV(II)}$, $t_{ROBI2}^{AGV(II)}$. On the basis of relation (13) one copy from $ROPN^{AGV}$ containing the input transition $t_{RIS}^{AGV(III)}$ and the output transition $t_{ROBI1}^{AGV(III)}$ is created. On the basis of relation (14) one copy from $ROPN^{AGV}$ containing the input transition $t_{RIBO2}^{AGV(IV)}$ and the output transition $t_{ROS}^{AGV(IV)}$ is created. Roman numerals in brackets identify resource cycle number. All four cycles start from p_{RId}^{AGV} place (Fig. 5).

The models of the buffer storages BI1, BI2, BO1 and BO2 are defined in the same way. In the next step the models of machine tools are defined. Taking as an example the MS1 machine a copy from $ROPN^{MS1}$ is created. It contains: the input transition $t_{RIBI1}^{MS1(I)}$ for the pair $(BI1, MS1) \in RI_{s1}^1$, the process transitions $t_{RP}^{MS1(I)}$ and $t_{RP}^{MS1(II)}$ which are counterparts of the transitions t_{PO1}^1 and t_{PO3}^1 representing the operations OA_1^1 and OA_3^1 respectively, the output transition $t_{ROBO1}^{MS1(I)}$ for the pair $(MS1, BO1) \in RI_{12}^1$, the output transition $t_{ROBO1}^{MS1(II)}$ for the pair $(MS1, BO1) \in RI_{34}^1$ (Fig. 6). According to the same rules the model of the MS2 machine is created. It contains two input transitions for the pairs $(BI2, MS2)$ in RI_{12}^1 and RI_{34}^1, two process transitions for the operations OA_2^1 and OA_4^1 and one output transition for the pair $(MS2, BO2)$ in RI_{2e}^1. Finally a model for the storage S is constructed. For the presented example $S = R^e = R^s$, so the model contains cycles for both R^e and R^s resources (Fig. 7). It contains input transitions: $t_{RIAGV}^{S(I)}$ for relation (11), $t_{RIAGV}^{S(II)}$ for relation (12), $t_{RIAGV}^{S(IV)}$ for relation (14) and output transitions: $t_{ROAGV}^{S(I)}$ for relation (11), $t_{ROAGV}^{S(II)}$ for relation (12), $t_{ROAGV}^{S(III)}$ for relation (13).

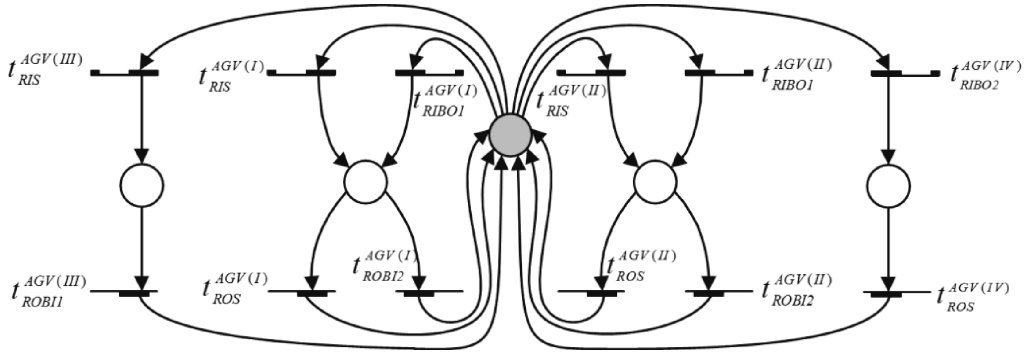

Fig. 5. A graphical view of operation cycles of a model for AGV.

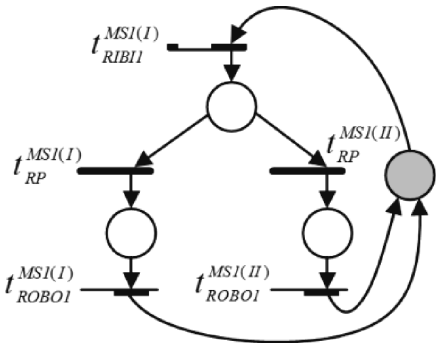

Fig. 6. A graphical view of operation cycles of a model for machine tool MS1.

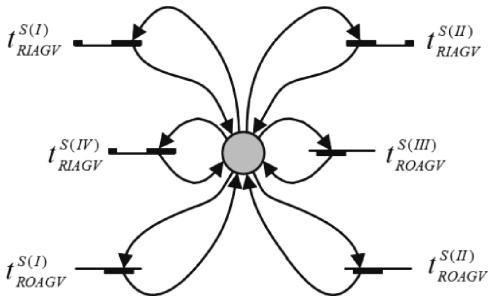

Fig. 7. A graphical view of operation cycles of a model for storage S.

The created models for resources are supplemented with parts' flow information. The places and flow relation defining the states of parts between the input and output transitions are added to the resource models. Places representing the parts added to the model of MS1 are marked by bold circles (Fig. 8). The model of storage S represents both the R^s and R^e resource. So, the model for S resource is supplemented with a place representing parts in an initial state before the output transition $t^{S(III)}_{ROAGV}$ and a place representing parts in the final state after the input transition $t^{S(IV)}_{RIAGV}$ (Fig. 9).

Input and output transitions in a resource model serve as integration interfaces. Models of all the resources are integrated by the definition pairs of input and output transitions according to the transition incidence relation (Fig. 10). For each model only the input and output transitions are shown and they are connected by arrows.

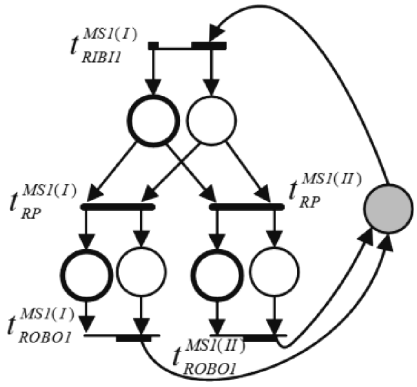

Fig. 8. A graphical view for machine tool MS1 model with states of processed parts.

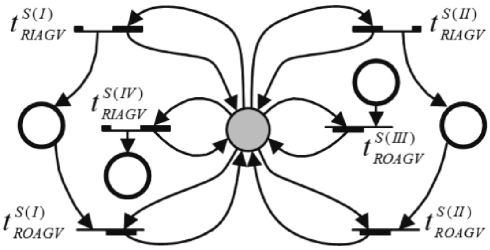

Fig. 9. A graphical view for storage S model with states of processed parts.

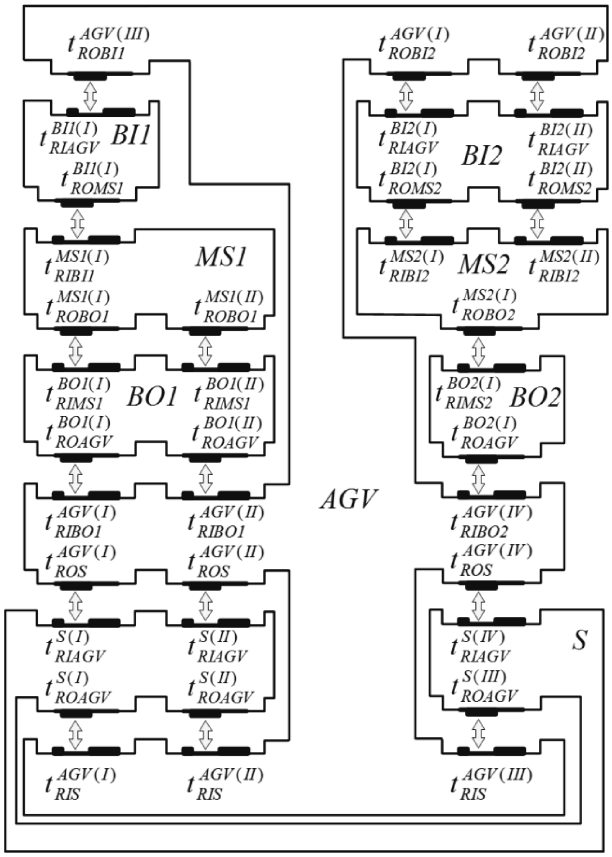

Fig. 10. A diagram of integrated models.

Conclusions

The generation of NLPP extend the robustness of an automated manufacturing control system with respect to internal and external disturbances. The main objective of the control system is the selection of optimal decisions, taking into consideration applied optimality criteria, current state of the manufacturing system and constraints that prevent DEMS from transition to an undesirable state like deadlock [34, 35] or starvation.

A proper way to build a control system, which works effectively in an unpredictable environment, is to prepare a model whose essential feature is integration of the data which define the NLPP and DEMS structure into one coherent model. The proposed model contains all the data about alternatives in process plans and also possible execution of alternative processes in the system. The model does not limit the decision space in any way and thus enables the control system to utilize flexible online scheduling (job rerouting) taking into account actual state of the system. Model generation algorithm, which is the main contribution of the paper, enables quick and automatic model reconfiguration in case new jobs are added for execution, some jobs are finished or modified.

The model may be used for both offline system simulation and scheduling as well as online control. Modular structure of the model makes it manageable even for large scale systems and suitable for promising agent based technology. Usage of widely recognized Petri net formalism simplifies model creation and understanding, moreover powerful tool for model verification are available off the shelf.

References

[1] Madki S.J., Pawar Dr.M.S., *Computer Aided Process Planning (CAPP) for Manufacturing in Job Type Industries- A Review*, International Journal of Emerging Technology and Advanced Engineering, 5, 7, 213–215, 2015.

[2] Duda J., *Semigenerative System for Manufacturing Process Planning*, Proceedings of the 10th FAIM International Conference, University of Maryland, USA, 2, 1007–1016, 26–28 June 2000.

[3] Sormaz D.N., Khoshnevis B., *Generation of alternative process plans in integrated manufacturing systems*, Journal of Intelligent Manufacturing, 14, 509–526, 2003.

[4] Weiming S., Lihui W., Qi H., *Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey*, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 36, 4, 563–577, 2006.

[5] Kiristis D., Porchet, A., *A generic Petri Net Model for Dynamic Process Planning and Sequence Optimization*, Advances in Engineering Software, 25, 1, 61–71, 1996.

[6] Capek R., Sucha P., Hanzalek Z., *Production scheduling with alternative process plans*, European Journal of Operational Research 217, pp. 300–311, 2012.

[7] Li X.Y., Zhang C.Y., Gao L., Li W.D., Shao X.Y., *An agent-based approach for integrated process planning and scheduling*, Expert Systems with Applications, 37, 1256–1264, 2010.

[8] Son Y.J., Wysk R.A., *Automatic simulation model generation for simulation-based, real-time shop floor control*, Computers in Industry, 45, 291–308, 2001.

[9] Li X., Gao L., Zhang C., Shao X., *A review on Integrated Process Planning and Scheduling*, Int. J. Manufacturing Research, 5, 2, 161–180, 2010.

[10] Phanden R.K., Jain A., Verma R., *Review on integration of process planning and scheduling*, DAAAM International Scientific Book, Vienna, Austria, pp. 593–618, 2011.

[11] Sachin S., Sanjay J., *Integration of Process Planning and Scheduling Comparison of Models Approach*, International Journal of Science and Research, 1, 3, 215–219, 2012.

[12] Kumar M., Rajotia S., *Integration of scheduling with computer aided process planning*, Journal of Materials Processing Technology, 138, 297–300, 2003.

[13] Gregor M., Skorik P., *Simulation and emulation of manufacturing systems behaviour*, Management and Production Engineering Review, 1, 2, 11–21, 2010.

[14] Ghasemi F., Momeni M., Sharifi A., Piran M., *A Simulation Model of Production Scheduling*, World Applied Sciences Journal, 18, 6, 861–867, 2012.

[15] Zajac J., Slota A., Chwajol G., *Distributed Manufacturing Control: Models and Software Implementations*, Management and Production Engineering Review, 1, 1, 38–56, 2010.

[16] Hanisch H.M., Kemper P., Lüder A., *A Modular and Compositional Approach to Modeling and Controller Verification of Manufacturing Systems*, Proceedings of 14th IFAC World Congress, vol. J, Beijing, China July, pp. 187–192, 1999.

[17] Masri A., Bourdeaud'huy T., Toguyeni A., *A component-based approach based on High-Level Petri Nets for modeling Distributed Control Systems*, International Journal on Advances in Intelligent Systems, 2, 2 & 3, 335–353, 2009.

[18] Booch G., Rumbaugh J., Jacobson I., *The Unified Modelling Language User Guide*, Addison Wesley 1999.

[19] Ismail H.S., Tey V.S., Wang L., Poolton J., *A UML Approach for the Design of Reconfigurable Manufacturing Simulation Models*, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, pp. 1690–1694, December 2011.

[20] Chandra V., Kumar R., *A New Modeling Formalism and Automata Model Generator for a Class of Discrete Event Systems*, Proceedings of The American Control Conference, Arlington, VA, pp. 4562–4567, June 2001.

[21] Banaszak Z. [Ed.], *Modelling and control of FMS. Petri net approach*, Wrocław Technical University Press, Wrocław 1991.

[22] Moore K.E., Gupta S.M., *Stochastic Colored Petri Net Models of Flexible Manufacturing Systems: Material Handling Systems and Machining*, Computers and Industrial Engineering, 29, 1–4, 333–337, 1995.

[23] Dicesare F., Harhalakis G., Proth J.M., Silva M., Vernadat F.B., *Practice of Petri Nets in Manufacturing*, Chapman & Hall, 1993.

[24] Brauer W., Reisig W., *Carl Adam Petri and Petri Nets*, Informatik-Spektrum, 29, 5, 369–374, 2006.

[25] Dotoli M., Fanti M., Giua A., Seatzu C., *First-order hybrid Petri nets. An application to distributed manufacturing systems*, Nonlinear Analysis: Hybrid Systems, 2, 2, 408–430, 2008.

[26] Julia S., de Oliveira F.F., Valette R., *Real time scheduling of workflow management systems based on a p-time petri net model with hybrid resources*, Simulation Modelling Practice and Theory, 16, 4, 462–482, 2008.

[27] Petrucci L., *Modularity and Petri Nets*, Proc. 7th Int. Symposium on Programming and Systems (ISPS'2005), Algiers, Algeria, pp. 7–8, May 2005.

[28] Tsinarakis G.J., Tsourveloudis N.C., Valavanis K.P., *Modular Petri Net based modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems*, Journal of Intelligent Manufacturing, 16, 1, 67–92, 2005.

[29] Lee H., Banerjee A., *A Modular Petri Net based Architecture to Model Manufacturing Systems Exhibiting Resource and Timing Uncertainties*, Proceedings of 5th Annual IEEE Conference on Automation Science and Engineering (CASE 2009), Bangalore, India, pp. 525–530, August 2009.

[30] Kiritsis D., Neuendorf K.P., Xiruchakis P., *Petri net techniques for process planning cost estimation*, Advances in Engineering Software, 30, 375–387, 1999.

[31] Cyklis J., Zajac J., Slota A., *Models of Manufacturing System for simulation and control*, Journal of Manufacturing Engineering, 3, 4, 10–15, 2004.

[32] Slota A., *Petri Net model of Flexible Assembly Systems for distributed control*, Advances in Manufacturing Science and Technology, 29, 4, 91–98, 2005.

[33] Habel J., *The idea of machining process planning with alternative routes in form of non-cyclic graph for CAPP implementation*, Technical Transactions Mechanics, 1-M/2013, pp. 145–153, 2013.

[34] Banaszak Z., Krogh B.H., *Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows*, IEEE Trans. on Robotics and Automation, 6, 6, 724–734, 1990.

[35] Zajac J., *A Deadlock Handling Method for Automated Manufacturing Systems*, CIRP Annals – Manufacturing Technology, 53, 1, 367–370, 2004.