

# Automatic Synthesis of Multiple Place Resource Models with Petri Nets

Luca Ferrarini, *Senior Member, IEEE*, and Luigi Piroddi  
 Dipartimento di Elettronica e Informazione  
 Politecnico di Milano  
 Piazza Leonardo da Vinci 32, I-20133 Milano (Italy)  
 {ferrarini,piroddi}@elet.polimi.it

**Abstract** – The paper focuses on the correct representation of shared resources in Petri net models of automated manufacturing systems, in the presence of concurrent independent operations, such as assembly and disassembly ones. In particular, an original multiple-place model is specifically derived and a constructive algorithm is proposed for the automatic synthesis of the resource model. The algorithm computes the number of resource places for each resource, as well as their connections to the recipes’ transitions. Proper necessary conditions of “well-definiteness” are introduced for the partial models that are incrementally defined by the user.

## I. INTRODUCTION

The recent years have witnessed a considerable effort in the development of Petri net based formal design methodologies for the modeling and control of automated manufacturing systems [1-7]. Particularly worth mentioning are the works [5-6], where a classic top-down approach for the specification of operations and recipes is complemented with an incremental approach for the inclusion of constraints and resources. Bottom-up approaches, based on the merging or aggregation of sub-nets, have also been considered. Typically, strongly connected state machines are employed in these approaches, see e.g. the Resource Control Nets developed in [1, 4] and the Elementary Control Tasks of [7].

In all these methodologies, system resources are usually represented with single places, with one or more tokens [4, 5, 8]. More specifically, the resource allocation is represented by arcs directed from the resource place to the begin transitions of the corresponding operations, while arcs directed from the end transitions of operations to the resource place are used to describe the resource release. Though adequate in many situations, such a modeling scheme presents some limitations in the case of complex concurrent operations. Consider for example a Petri net model representing the synchronization between two operations  $o_1$  and  $o_2$ , followed by a third operation  $o_3$  (Fig. 1.a). The three operations require the use of the same resource  $r_1$  and  $o_1$  and  $o_2$  can be executed independently. A typical single place resource model would look as the one in Fig. 1.b, where  $k$  is the marking of the resource place and  $\alpha$ ,  $\beta$ ,  $\gamma$  are the arcs’ weights. The net of Fig. 1.b is easily shown to be not live, regardless of the values of  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $k$  (assume that  $\alpha \leq \beta$ ; then a repeated execution of  $o_1$  will eventually result in a deadlock state). To avoid this, either the representation of  $o_1$  and  $o_2$  has to be simplified, e.g. representing them as a unique joint operation, or further constraints must be introduced in the model to impose a

correct behavior of the resource allocation mechanism, e.g. forcing an actuation order for  $o_1$  and  $o_2$ .

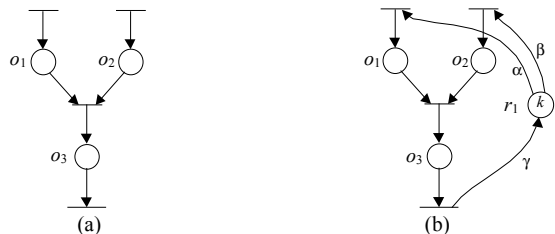


Fig. 1 Resource allocation example

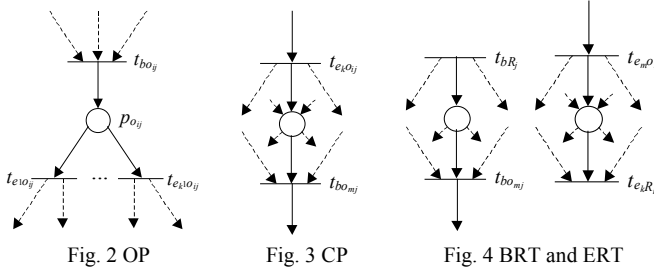
This example is representative of relevant modeling problems, such as complex assembly/disassembly operations involving different parts from separate *independent* processes, common in manufacturing systems. The present paper introduces a modeling technique for the representation of shared resources, which extends the classical scheme previously recalled, and allows the modeling of complex resource allocation mechanisms. Multiple places may be required to model a single resource. A constructive algorithm is proposed (Sect. III) for the automatic synthesis of a correct resource model. In order to reject unfeasible requests, the concept of *well-defined resource model* is introduced (Sect. IV). Briefly, the algorithm ensures that given a correctly defined recipe, each added resource constraint (considered individually) preserves the net’s basic properties. Notice that this guarantees only the representation correctness of the allocation/release mechanism for each resource, while it does not automatically avoid deadlocks resulting from circular wait conditions involving two or more resources. For the last purpose, the application of a deadlock prevention algorithm is mandatory. An FMS example is finally discussed in Sect. V to validate the approach.

## II. MODELING OF OPERATIONS AND RECIPES

Let  $S_O = \{o_1, o_2, \dots, o_{N_O}\}$  be the set of  $N_O$  operations performed by the process to be modeled. Each operation must use at least one of the  $N_r$  resources in the set  $S_r = \{r_1, r_2, \dots, r_{N_r}\}$ . For each resource, the number of available units (resource capacity) is also specified. Some resources may be *shared* among different operations, giving rise to possible deadlock situations. Notice that a resource may be used by consecutive operations, without intermediate release. To specify complete production tasks, involving several, suitably connected and synchronized operations, a set  $S_R = \{R_1, R_2, \dots, R_{N_R}\}$  of  $N_R$  recipes is also introduced. Only recipes without recirculating paths are considered in the present

work. A formal definition of these specifications will be developed in the following using Petri nets (PN) (see e.g. the review paper [9] for details on PN's). The PN model for  $R_j$  will be denoted  $N_{R_j}$ .

Similarly to what is done in [3], each operation  $o_i$  used in the recipe  $R_j$  is modeled with an *operation place* (OP)  $p_{o_{ij}}$  (marked when operation  $o_i$  is in execution), a *begin operation transition* (BOT)  $t_{bo_{ij}}$  and  $k_i$  *end operation transitions* (EOT)  $t_{e_{1o_{ij}}}, \dots, t_{e_{k_i o_{ij}}}$  (see Fig. 2). Extension to multiple BOT's is straightforward. Sequential connections are realized by means of *connection places* (CP), which model the intermediate state in a sequence of two operations, as depicted in Fig. 3. Connection places are introduced to preserve the one-transition-one-event paradigm, though the resulting models can be shown to be completely equivalent to classical ones with operation and resource places only. To account for the “beginning” and “end” of recipe  $R_j$ , a *begin recipe transition* (BRT)  $t_{bR_j}$  and  $k_j$  *end recipe transitions* (ERT)  $t_{e_{1R_j}}, \dots, t_{e_{k_j R_j}}$  must be added to the recipe model (see Fig. 4). Places in Fig. 4 are also CP's.



The above modeling rules represent mere constraints for the representation of connection structures, and do not guarantee the correctness of the recipe model. For this reason, a “well-definiteness” condition must be introduced for the recipe model. For this purpose, with reference to a recipe model  $N_{R_j}$ , consider the augmented net  $N_{R_j}'$ , obtained by adding an *augmentation place* (AP)  $p_{0j}$ , as shown in Fig. 5.

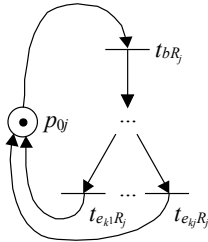


Fig. 5 Augmented recipe model

**Definition 1 — Well-defined recipe model**

A recipe model  $N_{R_j}$  is said to be *well-defined* if its augmented version  $N_{R_j}'$  is live and safe.  $\square$

**Lemma 1 —** Let  $N_{R_j}$  be a well-defined recipe model. Then, its augmented version  $N_{R_j}'$  is reversible.  $\square$

*Proof.* Initially, only  $t_{bR_j}$  is enabled, and the AP is emptied when it fires. By the liveness assumption, the AP can

always be marked again from any reachable state (with one token only, since  $N_{R_j}'$  is safe). Suppose, *ab absurdo*, that not all OP's and CP's are left unmarked when this happens. Then, the reached marking would be strictly greater than the initial marking and the net would be unbounded.  $\square$

**Definition 2 — Recipe execution sequence**

With reference to a recipe model  $N_{R_j}$  a firing sequence starting with  $t_{bR_j}$ , ending with a  $t_{e_{k_j R_j}}$  and containing only one  $t_{bR_j}$  is termed a *recipe execution sequence* (RES).  $\square$

The absence of deadlocks in  $N_{R_j}$  is necessary to guarantee the possibility to complete the recipe. Reversibility is also a necessary property to ensure that the initial state of the recipe model can always be recovered after a RES. Finally, boundedness implies that there cannot be reachable states with a marking strictly greater than that of the initial state (with only the AP marked), which would allow the execution of operations when the recipe is inactive.

**Lemma 2 —** Let  $N_{R_j}$  be a well-defined recipe model.

Then, all its transitions fire at most once in a RES.  $\square$

*Proof.* Suppose, *ab absurdo*, that transition  $t$  can fire twice. In the absence of recirculating paths, the enabling of  $t$  does not depend on the marking of the places in  $t \bullet$ . Then, there exists a reachable marking in which the same places are marked with two tokens, violating the safeness assumption on  $N_{R_j}'$ .  $\square$

The *overall recipe model*  $N_R$  is the composition of all the recipe models,  $N_R = N_{R_1} \circ \dots \circ N_{R_{N_R}}$ , where the symbol ‘ $\circ$ ’ denotes the transition synchronization operator.

### III. MODELING OF RESOURCES

Resource usage is typically modeled by introducing one *resource place* (RP) for each resource, initially marked with the resource availability, and connected with outgoing [incoming] arcs to BOT's [EOT's] associated to operations using the resource [8]. If the resource is to be retained between subsequent operations, only the first BOT and the last EOT of the sequence are connected to the RP. However, multiple RP's may be needed to describe the concurrent usage of a single resource by different operations, as in assembly/disassembly operations. For example, resource  $r_1$  in Fig. 6b has been modeled with two RP's,  $p_{r_{11}}$  and  $p_{r_{12}}$ , since it is used in concurrent mode by two independent parallel branches. Notice that resource  $r_3$  in Fig. 6a is not modeled correctly, because it is allocated before an “alternative” connection, but released only in one of the alternative branches. In Fig. 6b, resource  $r_4$  is not modeled correctly, since the RP  $p_{r_4}$  turns out to be unbounded.

A resource that is held throughout the execution of multiple operations, from a transition  $t_{bo_i}$  to a transition  $t_{e_{km}o_m}$ , must be used in all elementary paths between the same

transitions, denoted  $ep(t_{bo_i}, t_{e_{km}o_m})$ . Therefore the notion of macro-operation is introduced.

**Definition 3 — Macro-operation**

With reference to a well-defined recipe model  $N_{R_j}$ , the set of all elementary paths from  $t_{bo_i}$  to  $t_{e_{km}o_m}$  (if any exist) defines a *macro-operation*  $M = \{ep(t_{bo_i}, t_{e_{km}o_m})^1, \dots, ep(t_{bo_i}, t_{e_{km}o_m})^p\}$ . The begin [end] transition  $t_{bo_i}$  [ $t_{e_{km}o_m}$ ] of  $M$  is denoted  $t_{bM}$  [ $t_{eM}$ ]. The *support*  $\|M\|$  of a macro-operation  $M$  is the set of all nodes belonging to elementary paths in  $M$ .  $\square$

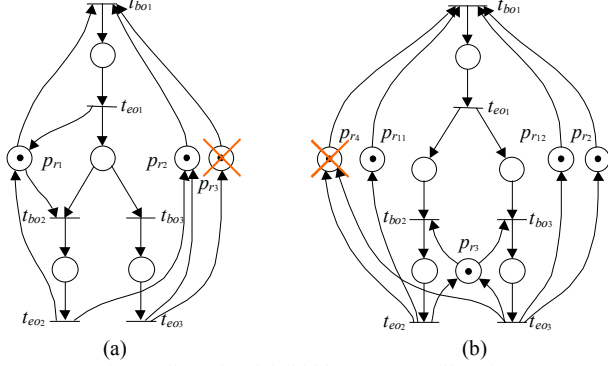


Fig. 6 Allowed and forbidden resource allocations

By definition, there cannot be two distinct macro-operations with the same initial and final transitions.

**Definition 4 — Alternative macro-operations**

With reference to a well-defined recipe model  $N_{R_j}$ , let  $M_1$  and  $M_2$  be two distinct macro-operations (i.e. either  $t_{bM_1} \neq t_{bM_2}$  or  $t_{eM_1} \neq t_{eM_2}$ , or both).  $M_1$  and  $M_2$  are said to be *alternative macro-operations* iff there does not exist any RES, which contains more than one transition in  $\{t_{bM_1}, t_{bM_2}\}$  and more than one transition in  $\{t_{eM_1}, t_{eM_2}\}$ .  $\square$

Notice that any two macro-operations  $M_1$  and  $M_2$  such that  $t_{eM_1}$  and  $t_{eM_2}$  are different EOT's of the same operation, are alternative by construction. It is important to group alternative macro-operations which are also partially overlapping, since an inconsistent model would be obtained if such macro-operations were not to use the same resource (see e.g. resource  $r_3$  in Fig. 6a). To this end, the concept of *complete set of alternative macro-operations* is introduced.

**Definition 5 — Complete set of alternative macro-op.s**

With reference to a well-defined recipe model  $N_{R_j}$ , a set  $S$  of alternative macro-operations is said to be *complete* iff:

- i. in any RES, the presence of a  $t_{bM_i}$  [ $t_{eM_i}$ ], with  $M_i \in S$ , implies the presence of one (and only one)  $t_{eM_j}$  [ $t_{bM_j}$ ], with  $M_j \in S$ ;
- ii. there does not exist a subset of  $S$  such that (i) holds.  $\square$

Def. 5 implies that in any RES either no  $t_{bM_i}$  nor  $t_{eM_i}$  are fired, for all  $M_i \in S$ , or the  $t_{bM_i}$  of *one* macro-operation  $M_i \in S$  and the  $t_{eM_j}$  of *one* macro-operation  $M_j \in S$  are fired, in this

order.

**Definition 6 — Independent macro-operations**

With reference to a well-defined overall recipe model  $N_R$ , let  $M_1$  and  $M_2$  be two macro-operations such that  $\|M_1\| \cap \|M_2\| = \emptyset$ .  $M_1$  and  $M_2$  are said to be *independent macro-operations* iff:

- i. there does not exist any  $ep(x_1, x_2)$  or  $ep(x_2, x_1)$ ,  $x_1 \in \|M_1\|$ ,  $x_2 \in \|M_2\|$ , or
- ii. the only paths between nodes of  $\|M_1\|$  and  $\|M_2\|$  either all contain  $ep(t_{eM_1}, t_{bM_2})$  or  $ep(t_{eM_2}, t_{bM_1})$ .  $\square$

The following lemma 3 proves that, if one of two independent macro-operations is started, its completion is not constrained by the activation of the other one.

**Lemma 3** — With reference to a well-defined overall recipe model  $N_R$ , let  $M_1$  and  $M_2$  be two independent macro-operations. Then, if  $t_{bM_1}$  is fired, the firing of transition  $t_{eM_1}$  is not constrained by the firing of  $t_{bM_2}$ .  $\square$

*Proof.* Suppose that  $M_1$  and  $M_2$  satisfy condition (i) of Def. 6. Then, it is not possible to modify the enabling condition of any transition in  $\|M_1\|$ , by changing the marking of places in  $\|M_2\|$  or by firing transitions in  $\|M_2\|$ . Suppose now that  $M_1$  and  $M_2$  satisfy condition (ii) of Def. 6. Then, if the only paths between nodes of  $\|M_1\|$  and  $\|M_2\|$  all contain  $ep(t_{eM_1}, t_{bM_2})$ , there do not exist paths from nodes of  $\|M_2\|$  to nodes of  $\|M_1\|$ , and  $t_{eM_1}$  is not constrained by the firing of  $t_{bM_2}$ . In the other case, the only paths between nodes of  $\|M_2\|$  and  $t_{eM_1}$  all contain  $t_{bM_1}$ , which is fired by assumption.  $\square$

Notice that two independent macro-operations  $M_1$  and  $M_2$  may be alternative. Independent macro-operations subject to condition (ii) of Def. 6 are strictly sequential. Consider now the problem of associating macro-operations to resources. With reference to the overall recipe model  $N_R$ , let  $M_{r_q}$  be the set of all macro-operations which require the resource  $r_q$  throughout the *whole* execution of their operations, such that there does not exist a macro-operation in the set, whose support is strictly contained in another one.

**Lemma 4** — Let  $M_1 \in M_{r_q}$ . Then  $\nexists M_2 \in M_{r_q}$  such that:

- i.  $t_{bM_2} \neq t_{bM_1}$  and  $t_{bM_2} \in \|M_1\|$ , or
- ii.  $t_{eM_2} \neq t_{eM_1}$  and  $t_{eM_2} \in \|M_1\|$ .  $\square$

*Proof.* Suppose, *ab absurdo*, that there is a macro-operation  $M_2$  such that condition (i) is verified. Then construct all the elementary paths  $ep(t_{bM_1}, t_{bM_2})$  given by the concatenation of all the elementary paths  $ep(t_{bM_1}, t_{bM_2})$  in  $M_1$ , with all the elementary paths  $ep(t_{bM_2}, t_{eM_2})$  in  $M_2$ . Such elementary paths  $ep(t_{bM_1}, t_{eM_2})$  require the resource  $r_q$  throughout their *whole* execution. Therefore, there would exist another macro-operation  $M_3 \in M_{r_q}$ , with  $t_{bM_3} = t_{bM_1}$  and  $t_{eM_3} = t_{eM_2}$ , such that  $\|M_3\| \supset \|M_2\|$ , which is not compatible with the definition of set  $M_{r_q}$ . A similar line of proof can be adopted to prove that

$\nexists M_2 \in M_{r_q}$  fulfilling condition (ii).  $\square$

Based on these newly introduced concepts the following *Resource Definition Algorithm* (RDA) can be devised to model the resource usage:

- i. With reference to the overall recipe model  $N_R$ , divide the set  $M_{r_q}$  in *all* (not necessarily disjunct) sub-sets  $U_{r_q}^n$ ,  $M_{r_q} = \bigcup_{n=1}^p U_{r_q}^n$ , such that each  $U_{r_q}^n$  is a complete set of alternative macro-operations.
- ii. Partition the set  $U_{r_q} = \{U_{r_q}^1, U_{r_q}^2, \dots, U_{r_q}^p\}$  in sub-sets  $W_{r_q}^u$  such that macro-operations belonging to different sub-sets  $W_{r_q}^u$  are independent, and mutually exclusive with respect to the usage of resource  $r_q$ .
- iii. Let  $N_{r_q} = \prod_{u=1}^t |W_{r_q}^u|$ , where  $|\cdot|$  is the cardinality operator. Let  $C_{r_q}^s$ ,  $s = 1, \dots, N_{r_q}$ , be the possible  $t$ -ples  $\langle U_{r_q}^{s1}, \dots, U_{r_q}^{st} \rangle$  of elements  $U_{r_q}^{su}$ , s.t.  $U_{r_q}^{su} \in W_{r_q}^u$ ,  $u = 1, \dots, t$ .
- iv. Represent resource  $r_q$  with  $N_{r_q}$  RP's  $p_{r_{qs}}$ ,  $s = 1, \dots, N_{r_q}$ , each initially marked with a number of tokens equal to the resource capacity. Connect  $p_{r_{qs}}$  with outgoing [incoming] arcs to all the initial BOT's [final EOT's] of the macro-operations belonging to the  $t$  elements  $U_{r_q}^{su}$ ,  $u = 1, \dots, t$ , in  $C_{r_q}^s$ ,  $s = 1, \dots, N_{r_q}$ .

Notice that a macro-operation cannot belong to different  $W_{r_q}^u$ , whereas it can belong to multiple  $U_{r_q}^n$ . Macro-operations belonging to different  $U_{r_q}^n$  in the same  $W_{r_q}^u$  can use concurrently the resource  $r_q$ . On the other hand, macro-operations in different  $W_{r_q}^u$  are in mutual exclusion with respect to  $r_q$  and therefore must be completely distinct. The effect of RDA are summarized in Def. 7.

*Definition 7 — Specification model for resource  $r_q$*

The model for resource  $r_q$  is described by an acyclic marked PN  $N_{r_q} = (P_q, T_q, W_q, m_q)$ , where:

- i.  $P_q = \{p_{r_{qs}}\}$ ,  $s = 1, \dots, N_{r_q}$ , being the set of  $N_{r_q}$  RP's associated to resource  $r_q$ ;
- ii.  $T_q = \{t \mid t = t_{bM_i} \forall M_i \in M_{r_q}\} \cup \{t \mid t = t_{eM_i} \forall M_i \in M_{r_q}\}$ ;
- iii.  $W_q(p_{r_{qs}}, t_{bM_i}) = 1$  iff  $M_i \in U_{r_q}^{su}$  and  $U_{r_q}^{su}$  is contained in  $C_{r_q}^s$ , where  $s = 1, \dots, N_{r_q}$ ,  $j = 1, \dots, N_R$ , otherwise  $W_q(p_{r_{qs}}, t_{bM_i}) = 0$ ;  $W_q(t_{eM_i}, p_{r_{qs}}) = 1$  iff  $M_i \in U_{r_q}^{su}$  and  $U_{r_q}^{su}$  is contained in  $C_{r_q}^s$ , where  $s = 1, \dots, N_{r_q}$ ,  $j = 1, \dots, N_R$ , otherwise  $W_q(t_{eM_i}, p_{r_{qs}}) = 0$ ;
- iv.  $m_q(p_{r_{qs}}) = c_{r_q}$ , where  $c_{r_q}$  is the capacity of  $r_q$ .  $\square$

The *overall resource model*  $N_r$  is obtained by collecting all the single resource models in a single net by means of transition fusion:  $N_r = N_{r_1} \circ N_{r_2} \circ \dots \circ N_{r_{N_r}}$ .

#### IV. WELL-DEFINED RESOURCE MODEL

A minimum requirement for the formal correctness of the resource model is that whenever a resource is allocated to a recipe for one or more operations, the recipe cannot end without releasing the resource.

*Definition 8 — Well-defined resource model*

Let  $N_R = N_{R_1} \circ \dots \circ N_{R_{N_R}}$  be an overall recipe model, where all the recipe models  $N_{R_j}$ ,  $j = 1, \dots, N_R$ , are well-defined, and  $N_R' = N_{R_1}' \circ \dots \circ N_{R_{N_R}}'$  its augmented version. A resource model  $N_{r_q}$  is said to be *well-defined* iff the composed model  $N_R' \circ N_{r_q}$ , with AP's and RP's initially marked with one token each, is live, safe and reversible.  $\square$

For example, the incorrect modeling of  $r_3$  in Fig. 6a would result in a deadlocked and non reversible sub-net, whereas place  $p_{r_4}$  is unbounded in Fig. 6b, so that safeness and reversibility would be lost.

*Lemma 5 —* Let  $N_R = N_{R_1} \circ \dots \circ N_{R_{N_R}}$  be an overall recipe model, where all the recipe models  $N_{R_j}$ ,  $j = 1, \dots, N_R$ , are well-defined. Let a resource model  $N_{r_q}$  be constructed for resource  $r_q$  according to RDA. Then:

- i. Any two distinct macro-operations  $M_k, M_l \in M_{r_q}$ , such that  $M_k, M_l \in U_{r_q}^m$ , are connected to the same RP's.
- ii. Any two distinct macro-operations  $M_k, M_l \in M_{r_q}$ , such that  $M_k \in U_{r_q}^m$ ,  $M_l \in U_{r_q}^n$  and  $U_{r_q}^m \in W_{r_q}^i$ ,  $U_{r_q}^n \in W_{r_q}^j$ ,  $i \neq j$ , are connected to at least one common RP.
- iii. Any two distinct macro-operations  $M_k, M_l \in M_{r_q}$ , such that  $M_k \in U_{r_q}^m$ ,  $M_l \in U_{r_q}^n$ ,  $m \neq n$ ,  $M_k, M_l \notin U_{r_q}^m \cap U_{r_q}^n$ , and  $U_{r_q}^m, U_{r_q}^n \in W_{r_q}^i$ , are not connected to any common RP.  $\square$

*Proof.*

- i. By construction, a RP is either connected with all the macro-operations in  $U_{r_q}^m$  or with none of them.
- ii. By construction, there exists at least one  $t$ -ple  $C_{r_q}^s$  which contains both  $U_{r_q}^m$  and  $U_{r_q}^n$ . Since each  $t$ -ple is associated with a RP, all the macro-operations belonging to  $U_{r_q}^m \cup U_{r_q}^n$  will be connected to it.
- iii. By construction, there does not exist any  $t$ -ple  $C_{r_q}^s$  which contains both  $U_{r_q}^m$  and  $U_{r_q}^n$ . Since each  $t$ -ple is associated with a different RP,  $U_{r_q}^m$  and  $U_{r_q}^n$  are associated to distinct sets of RP's.  $\square$

*Lemma 6 —* Let  $N_R = N_{R_1} \circ \dots \circ N_{R_{N_R}}$  be an overall recipe model, where all the recipes are well-defined. Let a resource model  $N_{r_q}$  be constructed for resource  $r_q$  according to RDA, with all RP's marked with one token each. Assume that one transition  $t$  fires, such that  $\bullet t \cap P_q \neq \emptyset$ . Then there exists at

least one transition  $\hat{t}$ , which can be fired after the firing of  $t$ , such that  $\hat{t} \circ P_q = \bullet t \circ P_q$ .  $\square$

*Proof.* Transition  $t$  may have two kinds of RP's in its pre-set. RP's with only  $t$  in their post-set do not introduce constraints with respect to other transitions. When transition  $t$  fires, no other transition is inhibited as far as such RP's are concerned. On the other hand, RP's with multiple transitions in their post-set realize structural and effective conflicts between them, i.e. at most one of them can fire when they are all enabled. When transition  $t$  fires, all the other conflicting transitions in  $(\bullet t \circ P_q) \bullet$  are inhibited. By Lemma 5 each of these transitions:

- i. is either alternative to  $t$ , since by Lemma 5.i both are starting transitions of macro-operations which belong to a complete set of alternative macro-operations,
- ii. or it is the starting transition of a macro-operation, which is independent with respect to all macro-operations starting with  $t$  (Lemma 5.ii).

In the first case, at most one of the two transitions can fire, so that the RP's do not introduce additional constraints with respect to the recipe. In the second case, by Lemma 3, if one of the two transitions is fired, the firing of the second one is not required to complete a macro-operation starting with the first one. In all cases the unmarking of the RP's in  $\bullet t \circ P_q$  due to the firing of  $t$  does not inhibit the firing of any transition which could prevent the completion of a macro-operation starting with  $t$ . Therefore, since the overall recipe model is live, a transition  $\hat{t}$  will eventually be fired, which corresponds to the ending transition of a macro-operation starting with  $t$ . By RDA,  $\hat{t} \circ P_q = \bullet t \circ P_q$ , so that when  $\hat{t}$  is fired the RP's in  $\bullet t \circ P_q$  will be marked again.  $\square$

### Theorem 1 — Well-defined resource model

Let  $N_R = N_{R_1} \circ \dots \circ N_{R_{N_R}}$  be an overall recipe model, where all the recipe models  $N_{R_j}$ ,  $j = 1, \dots, N_R$ , are well-defined. Let a resource model  $N_{r_q}$  be constructed for resource  $r_q$  according to RDA. Then  $N_{r_q}$  is *well-defined*.  $\square$

*Proof.* Consider the net  $N_R \circ N_{r_q}$  initially marked with one token in each RP and AP. With reference to a generic reachable marking  $m$  in  $N_R \circ N_{r_q}$ , either all RP's are marked, and then there is *always* at least one enabled transition (since  $N_R$  is live, from the well-definiteness of  $N_{R_j}$ ,  $j = 1, \dots, N_R$ ), or some of them are unmarked. In the latter case, there are necessarily some macro-operations in execution using resource  $r_q$ . However, by Lemma 6, these macro-operations can always complete their execution, thereby restoring the original marking of the involved RP's. Since only the definitive unmarking of RP's may impair the original liveness of  $N_R$ , this proves that  $N_R \circ N_{r_q}$  is live. By Lemma 6, every time a RP is unmarked, its token will eventually be restored, so that RP's are safe. Then, since  $N_R$  is safe, also  $N_R \circ N_{r_q}$  is safe. Since  $N_R \circ N_{r_q}$  is live and there are not any recirculating paths in the net, then all activated recipes can indepen-

dently terminate by firing the corresponding ERT's. In the resulting state, the AP's are marked again. Since in  $N_R$  the AP's can be marked only when OP's and CP's are unmarked, and since all RP's are marked (no operation is active and using the resource  $r_q$ ), the initial marking  $m_0$  is restored. This proves the reversibility of  $N_R \circ N_{r_q}$ .  $\square$

The well-definiteness of the resource model has been proved in the worst case in terms of resource availability (RP's marked with only one token each). The preceding results can be simply extended to the case of multiple capacity resource places. In particular,  $N_R \circ N_{r_q}$  can be proved to be live, bounded and reversible.

## V. AN ILLUSTRATIVE EXAMPLE

Consider the flexible assembling/disassembling station represented in Fig. 7, which executes two different recipes  $R_1$  (disassembly) and  $R_2$  (assembly) (see Tab. I), involving 5 conveyor belts  $CB_i$ ,  $i = 1, \dots, 5$ , and two robots RBT1 and RBT2. Loading from CB2 and CB3 can be performed concurrently, as well as unloading to CB4 and CB5. The assembling/disassembling operations are in mutual exclusion.

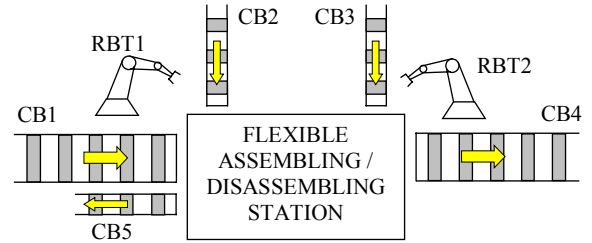


Fig. 7 The flexible assembling/disassembling station.

TABLE I RECIPES IN THE FLEXIBLE STATION EXAMPLE

recipe	operation	operation description
$R_1$	$o_1$	loading product from CB1 (with RBT1)
	$o_2$	disassembling product
	$o_3$	unloading part on CB5 (with RBT1)
	$o_4$	unloading part on CB4 (with RBT2)
$R_2$	$o_5$	loading part from CB2 (with RBT1)
	$o_6$	loading part from CB3 (with RBT2)
	$o_7$	assembling parts
	$o_8$	unloading product on CB4 (with RBT2)

An overall recipe model  $N_R$  is built according to the informal process specifications described above. The recipe nets are represented in dotted boxes in Fig. 8, where the 8 operation sub-nets have been integrated with 12 CP's and the recipe scheduling transitions ( $t_{bR_1}$ ,  $t_{eR_1}$ ,  $t_{bR_2}$ ,  $t_{eR_2}$ ). Note that operations  $o_3$  and  $o_4$  are performed in parallel, as well as  $o_5$  and  $o_6$ . The recipe models  $N_{R_1}$  and  $N_{R_2}$  are well-defined, since their augmented versions are live, safe and reversible (Def. 1 and Lemma 1). Overlooking for brevity reasons the role of the conveyor belts, three resources can be identified in the system, namely the two robot manipulators ( $r_1$  and  $r_2$ ) and the station itself ( $r_3$ ). The macro-operations using  $r_1$  are  $M_1$ ,  $M_3$  and  $M_5$ , with  $t_{bM_1} = t_{b0_1}$ ,  $t_{eM_1} = t_{e0_1}$ ,  $t_{bM_3} = t_{b0_3}$ ,  $t_{eM_3} = t_{e0_3}$ ,  $t_{bM_5} = t_{b0_5}$ ,  $t_{eM_5} = t_{e0_5}$ . Since

these are not alternative macro-operations,  $U_{r_1}^1 = \{M_1\}$  and  $U_{r_1}^2 = \{M_3\}$  and  $U_{r_1}^3 = \{M_5\}$ . In addition, these macro-operations are independent and mutually exclusive, so that  $W_{r_1}^1 = \{U_{r_1}^1\}$ ,  $W_{r_1}^2 = \{U_{r_1}^2\}$ ,  $W_{r_1}^3 = \{U_{r_1}^3\}$  and  $N_{r_1} = 1$ . The only possible 3-ple of elements  $U_{r_1}^n$  is  $C_{r_1}^1 = \langle U_{r_1}^1, U_{r_1}^2, U_{r_1}^3 \rangle$ . In view of this, one RP only must be introduced with outgoing arcs to transitions  $t_{bo1}$ ,  $t_{bo3}$  and  $t_{bo5}$  and incoming arcs from transitions  $t_{eo1}$ ,  $t_{eo3}$  and  $t_{eo5}$ . A similar resource model is computed for  $r_2$ , which uses in mutual exclusion the macro-operations  $M_4$ ,  $M_6$  and  $M_8$ , with  $t_{bM_4} = t_{bo4}$ ,  $t_{eM_4} = t_{eo4}$ ,  $t_{bM_6} = t_{bo6}$ ,  $t_{eM_6} = t_{eo6}$ ,  $t_{bM_8} = t_{bo8}$ ,  $t_{eM_8} = t_{eo8}$ . Resource  $r_3$  is used by the macro-operations  $M_{13}$ ,  $M_{14}$ ,  $M_{58}$  and  $M_{68}$ , with  $t_{bM_{13}} = t_{bo1}$ ,  $t_{eM_{13}} = t_{eo3}$ ,  $t_{bM_{14}} = t_{bo1}$ ,  $t_{eM_{14}} = t_{eo4}$ ,  $t_{bM_{58}} = t_{bo5}$ ,  $t_{eM_{58}} = t_{eo8}$ ,  $t_{bM_{68}} = t_{bo6}$ ,  $t_{eM_{68}} = t_{eo8}$ . Notice that  $M_{13}$  and  $M_{14}$  can be executed concurrently, as well as  $M_{58}$  and  $M_{68}$ , whereas the usage of  $r_3$  is mutually exclusive between  $M_{1j}$  and  $M_{k8}$  for  $j = 3, 4$ ,  $k = 5, 6$ . These macro-operations are not alternative macro-operations, so that  $U_{r_3}^1 = \{M_{13}\}$ ,  $U_{r_3}^2 = \{M_{14}\}$ ,  $U_{r_3}^3 = \{M_{58}\}$ ,  $U_{r_3}^4 = \{M_{68}\}$ . Now,  $M_{13}$  and  $M_{14}$  are partially overlapping and therefore not independent. The same applies to  $M_{58}$  and  $M_{68}$ . In view of this, the set  $U_{r_3}$  is partitioned in 2 sets  $W_{r_3}^1 = \{U_{r_3}^1, U_{r_3}^2\}$ ,  $W_{r_3}^2 = \{U_{r_3}^3, U_{r_3}^4\}$ . Since  $N_{r_3} = |W_{r_3}^1| \times |W_{r_3}^2| = 2 \times 2 = 4$ , four RP's must be introduced connected with the relative transitions (see Tab. II). The complete specification model is represented in Fig. 8.

TABLE II  $C_{R_3}^S$  2-PLS FOR THE FLEXIBLE STATION EXAMPLE

$C_{r_3}^s$	$U_{r_1}^n$	BOT's	EOT's
$C_{r_3}^1$	$\langle U_{r_3}^1, U_{r_3}^2 \rangle$	$t_{bo1}, t_{bo5}$	$t_{eo3}, t_{eo8}$
$C_{r_3}^2$	$\langle U_{r_3}^1, U_{r_3}^3 \rangle$	$t_{bo1}, t_{bo6}$	$t_{eo3}, t_{eo8}$
$C_{r_3}^3$	$\langle U_{r_3}^2, U_{r_3}^3 \rangle$	$t_{bo1}, t_{bo5}$	$t_{eo4}, t_{eo8}$
$C_{r_3}^4$	$\langle U_{r_3}^2, U_{r_3}^4 \rangle$	$t_{bo1}, t_{bo6}$	$t_{eo4}, t_{eo8}$

## VI. CONCLUSIONS

In the paper, the modeling of concurrently used shared resources in a generic manufacturing system is considered, such as arises in complex assembly and disassembly operations. An algorithm is proposed which computes the number of resource places and their connection to the recipes' transitions, on the basis of simple specifications in terms of resource usage. The specification model obtained by means of the presented methodology can be conveniently incorporated in the framework of a modular and hierarchical control structure, as hinted in [3]. This will give rise to a design methodology for Petri net models of manufacturing systems. Basic necessary conditions are provided for the well-definiteness of the partial models obtained, though clearly the liveness of the overall specification system cannot be proven by construction. Current research is focussing on the development of algorithms and rules for the construction of the recipe model, starting from high level prescriptions.

Also, the relationship between the well-definiteness conditions of the specification model and liveness enforcing algorithms will be investigated.

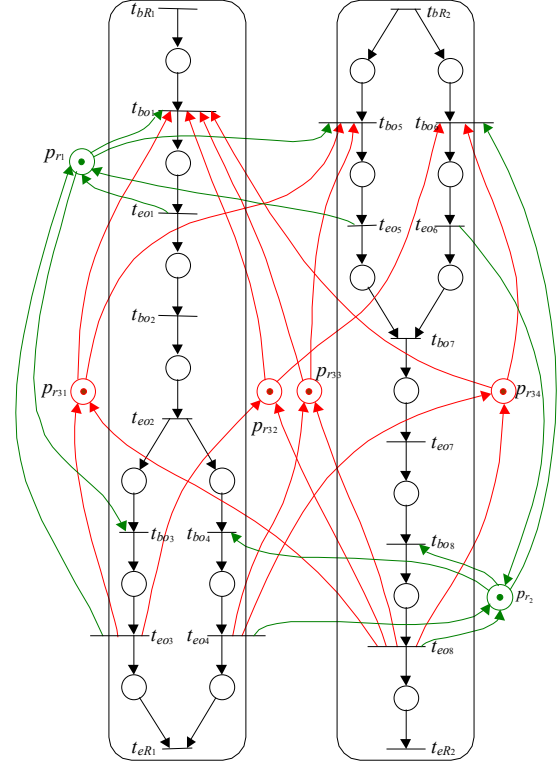


Fig. 8 Specification model for the flexible assembly/disassembly station

## REFERENCES

- [1] M.D. Jeng, "A Petri Net Synthesis Theory for Modeling Flexible Manufacturing Systems," *IEEE Trans on Systems, Man, and Cybernetics, part b*, Vol. 27, n. 2, 1997, pp. 169-183.
- [2] P. Falkman, B. Lennartson, and M. Tittus, "Modeling and Specification of Discrete Event Systems using Combined Process Algebra and Petri Nets," *IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics, AIM 2001*, Como (Italy), 2001, pp. 1011-1016.
- [3] L. Ferrarini, and L. Piroddi, "Design and implementation of a modular supervisory control system of a batch process," *European Control Conference, ECC 2001*, Porto (Portugal), 2001, pp. 1583-1588.
- [4] M.D. Jeng, and F. DiCesare, "Synthesis Using Resource Control Nets for Modeling Shared-Resource Systems," *IEEE Trans. on Robotics and Automation*, Vol. 11, n. 3, 1995, pp. 317-327.
- [5] M.C. Zhou, and F. DiCesare, "Parallel and Sequential Mutual Exclusions for Petri Net Modeling of Manufacturing Systems with Shared Resources," *IEEE Trans. on Robotics and Automation*, Vol. 7, n. 4, 1991, pp. 515-527.
- [6] M.C. Zhou, F. DiCesare, and A. Desrochers, "Hybrid Methodology for Synthesis of Petri Nets Models for Manufacturing Systems," *IEEE Trans. on Robotics and Automation*, Vol. 8, n.3, 1992, pp. 350-361.
- [7] L. Ferrarini, M. Narduzzi, and M. Tassan-Solet, "A new approach to modular liveness analysis conceived for large logic controller's design," *IEEE Trans. on Robotics and Automation*, Vol. 10, n. 2, 1994, pp.169-184.
- [8] M.C. Zhou, and F. DiCesare, "Petri Net Modeling of Buffers in Automated Manufacturing Systems," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 26, n. 1, 1996, pp. 157-164.
- [9] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings IEEE*, Vol. 77, n. 4, 1989, pp. 541-580.