Systematic Construction of Functional Abstractions of Petri Net Models of Typical Components of Flexible Manufacturing Systems

R. Z. Zurawski

Dept. of Electrical & Computer Eng.
Swinburne Inst. of Tech.
Melbourne, 3122
Australia

T. S. Dillon

La Trobe University
Dept. of Computer Science
Melbourne, 3083
Australia

Abstract

The use of generic models in the synthesis of FMSs, which allows for rapid modelling and analysis, does not reduce complexity of the verification model. This complexity can be reduced by replacing the constituent components (generic models) with their functional abstractions, which represent the external behaviour of the components. In this paper, for a class of Petri net models representing primary components of FMSs, a method that allows to systematically construct functional abstractions is presented. This method is then used to obtain the functional abstraction of the Petri net model of a machining station.

1. Introduction

Complex Petri net models of large manufacturing systems are inherently difficult to verify. For this reason a number of approaches, which attempt to reduce the difficulty of the verification task, were proposed in the past few years. Some of these approaches are based on bottom-up [1], [2], [3], top-down [4], [5], [6], [7], [8], [9], [10], [11] and structural reduction techniques. The top-down, bottom-up and structural reduction approaches, while reducing complexity of the verification task, may restrict the modelling flexibility, reuse of the submodels and the size of the synthesized net. An alternative approach that is free from all these modelling problems and reduces the verification task complexity is based on modelling with generic models and their functional abstractions. This approach provides the designer with Petri net models that represent typical components of production systems [12], [13], or even complete production system scenarios [14], thus allowing for rapid modelling and analysis. The use of generic models, however, does not reduce complexity of the verification model. Hence, the verification task of complex nets may be computationally and intellectually demanding due to a large number of places and transitions. In [15] a new approach to reduction of the complexity of the formal verification model is proposed. This approach is based on replacing the constituent components (generic models) of the verification model with their functional abstractions, which represent the external behaviour of these components. Since the objective of the introduction of functional abstractions is to reduce complexity of the verification model, the net realising the required external behaviour of a generic model should be as simple as possible. In the extreme case, the abstraction can be represented by a single transition or place. By replacing, in the verification model, complex subnets by simple ones, the number of places and transitions can be substantially reduced. Consequently the number of place (transition) invariants, and places (transitions) in the corresponding invariant supports, and the size of the reachability set can be substantially reduced as well. As a result, the verification task difficulty is reduced. The verification model, which is assembled from functional abstractions, is typically used to verify the interactions between the constituent components. Hence, the verification effort can primarily concentrate on the model structure and not on the internal nature of its components.

Temporal Petri nets are used in this paper to represent the external functionality of the Petri net models. The objective of this paper is to present a systematic approach to the construction of functional abstractions of a class of Petri net models which represent typical components of FMSs. This work uses place transition Petri nets as the specification language. Because of their simplicity, place transition Petri nets can be used as a powerful visual communication medium between the specification engineer and the

TH0386-3/91/0000/0248/$01.00 © 1991 IEEE

client. The ability of the client to understand the specification model is crucial to the system development time and cost.

The structure of the generic models of the primary components of FMSs, and properties of the corresponding functional abstractions are discussed in sections 2 and 3, respectively. Temporal Petri nets are introduced in section 4. Section 5 discusses the application of some of the structural techniques to the construction of functional abstractions. A method for systematic construction of functional abstractions is proposed in section 6. This method is used, in section 7, to construct the functional abstraction of the Petri net model of the machining station. Finally, section 8 provides the concluding remarks and suggestions for the future work.

2. Description of Petri Net Models of Primary Components

In order to propose a method for systematic construction of functional abstractions of Petri net models of primary components of FMSs, we restricted our considerations, and consequently the scope of the applicability of this method, to the class of Petri net models which have been proposed in [12] and [16].

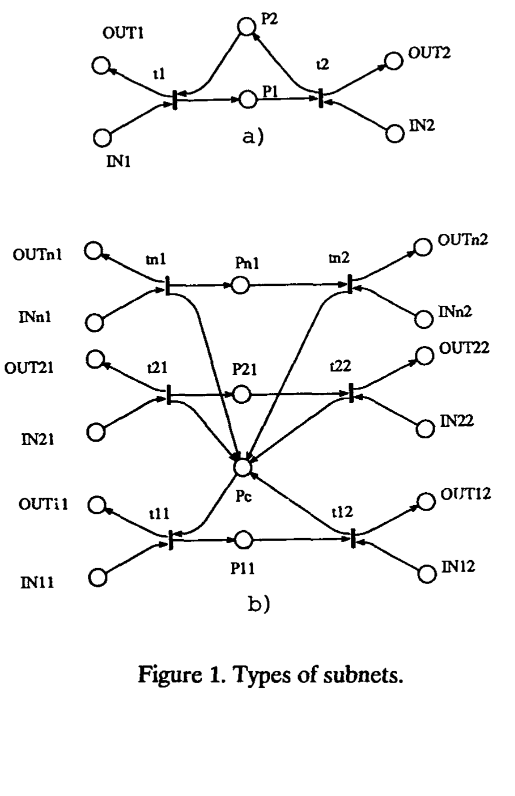

Figure 1. Types of subnets.

This class includes two types of models; namely models which are composed of one or more subnets of the kind shown in figure 1a, and models which are composed of one or more subnets of the kind shown in figure 1b. The subnets are combined into nets by the sharing of common transitions. The input and output interface places (places via which the model interacts with its environment) are optional. The first type of models were used to describe the primary components which can handle one part type. The second type of models were used to describe the primary components which can handle n different part types. In place transition Petri nets tokens have no identity. This identity, however, is required to distinguish amongst different types of parts. Thus, in order to obtain a Petri net model of a primary component that can handle n part types, n Petri net models of the primary component that can handle only one part type can be combined into a single net. This achieved by the sharing of common places P_C. In this paper, however, we concentrate on the first type of model. The results obtained in section 6 are valid also for the second type of model. Thus, the proposed approach to the systematic construction of functional abstractions can be easily extended to the second type of models.

Petri net models of the primary components represent flows of two types of objects. These are atomic and compound objects. The atomic objects represent physical resources, such as carts, pallets, etc., and control information. Flows of atomic objects, which represent physical resources, are modelled by one or more separate paths. A path always begins with an input interface place and ends with an output interface place, and does not contain any loops. Flows of the atomic objects which represent control information are modelled by two types of paths; paths that begin and end with a transition, and paths that begin with a transition and end with an output interface place. The former type represents the flow of the control information which is internal to the model. Each place that belongs to the path of this type has one input and one output transition only. The later type describes the flow of the control information which represents, in general, a request for service. This control information is exported to the model environment; it will be termed export control information. Paths of this type are, typically, modelled by one transition, one place (output interface place) and a directed arc.

The compound objects involve two or more atomic objects. An example of the compound object is the export control information-physical resource. For the class of Petri net models discussed in this paper, the path of flow of the compound object consists of two separate sections.

3 Properties of Functional Abstractions

Functional abstractions of the Petri net models used to describe the primary components of FMSs should exhibit the following properties:

The functional abstraction of a Petri net model should include all interface places of this model. The reason is that the interaction between the model and its environment is achieved via these places. The interface places are also the terminating places of the paths of flow of the physical resources and control information which represent requests for service. These paths provide connections between the input and output interface places, and are essential for the correct implementation of the external functionality of the model.

The abstraction should realize the external functionality of the model. This external functionality can be described by the input-output relationship; the presence of tokens on the input interface places should result in the presence of tokens on the output interface places of the model. Two types of the input-output relationships can be distinguished for Petri net models of the primary components. These two types of models are models with a single input interface place and a number of output interface places, and models with multiple input and output interface places. In the former case, tokens appear on the output places, in a fixed order, in response to the presence of a token on the input interface place. In the later case, tokens appear on the output interface places, also in a fixed order, in response to the presence of tokens on the input interface places, provided that tokens appear on the input interface places in a proper order.

Functional abstractions should provide the continuity of flows of objects that represent physical resources and control information. The use of structural analysis for verification of models, which are assembled from functional abstractions, requires the paths of flow of the objects to be closed. In Petri net models proposed in [12], the need to close paths is associated with the flow of the export control information-physical resource; this compound object represents a request for service and the subsequent response of the environment of the model.

Petri nets representing functional abstractions should be logically correct; otherwise they are useless. Functional abstractions are assembled into verification models in order to study correctness of the interactions between the constituent components of the model of a system.

4. Applications of Temporal Petri Nets to Modelling External Functionality of Primary Components of FMSs.

The need for functional abstractions to realize the external functionality of Petri net models, requires that the external functionality is expressed in an unambiguous and formal way. The formal description of the required external functionality can be used as a template against which the external functionality of a functional abstraction can be validated. The external functionality of a model can be specified using temporal Petri nets [17], [18]. Temporal Petri nets allow one to express temporal assertions on the behaviour of the model's interface without the need to define its internal implementation.

The temporal Petri net is a pair TPN = (PN, f).

Where PN is a Petri net:

PN = (P, T, IN, OUT) where

$P = \{p_1, p_2, ..., p_n\}$ is a set of places

$T = \{t_1, t_2, ..., t_m\}$ is a set of transitions

$P \cap T = \phi, P \cup T \neq \phi$

IN:(PxT) − > N is an input function of PN that defines directed arcs from place to transition. N is a set of non-negative integers.

OUT:(PxT) − > N is an output function of PN that defines directed arcs from transition to place.

$IP(t_j) = \{p_i \in P: IN(p_i, t_j) \neq 0\} \; \forall \; t_j \in T$; $IP(t_j)$ is the set of input places of t_j.

$OP(t_j) = \{p_i \in P: OUT(p_i, t_j) \neq 0\} \; \forall \; t_j \in T$; $OP(t_j)$ is the set of output places of t_j.

And f is a formula having the following syntax:

1) Propositions: p, t_{fir}, and t , where $p \in P$ and $t \in T$, are atomic propositions.
2) Atomic propositions are formulas.
3) If f and g are formulas then so are $\neg g, f + g, f \cdot g, f => g$, of ,$\Box f$, $\Diamond f$.

The atomic propositions p, t_{fir} and t mean that there is at least one token on place p in the current marking, transition t is firable in the current marking and transition t fires in the current marking, respectively. Symbols $\neg$, +, $\cdot$ and => represent the Boolean connectives. The formula of, " next ", means that f becomes true in the next marking. The formula $\Box$f, " henceforth", means that f becomes true in every marking reached from the current marking. The formula $\Diamond$f, "eventually", means that f becomes true at some marking reachable from the current marking.

The formal semantics of TPN is given as follows. Let $T^{oo} = T^* \bigcup T^{\omega}$, where T^{oo} is the set of all (finite and infinite) sequences of elements of T. T^* is the set of all finite sequences of elements of T, including the empty element λ. T^W is the set of all infinite sequences of elememts of T. Let α, β be sequences of elements of T. $|\alpha|$, $|\beta|$ denote the length of α, $\beta \in T^*$. The length $|\alpha|$, $|\beta|$ of $\alpha, \beta \in T^{\omega}$ is denoted by ω, where $i<\omega$ for every integer i. The concatenation of α and β, $\alpha\beta$, is finite if both sequences are finite, i.e., $\alpha, \beta \in T^*$; infinite if one sequence is infinite, i.e., $\alpha \in T^*$, $\beta \in T^{\omega}$; not defined if both sequences are infinite, i.e., $\alpha, \beta \in T^{\omega}$. Let L(PN, M), where M is a marking of PN, be a set of all finite firing sequences from M, L^{ω}(PN, M) a set of all infinite firing sequences from M, and L^{oo}(PN,M) a set of all (infinite and finite) firing sequences from M. Then $L^{oo}(PN, M) = L(PN, M) \bigcup L^{\omega}(PN, M)$. Let M be a marking of PN, and $\alpha \in L^{oo}(PN,M)$ is a possibly infinite firing sequence from M. Let $\alpha = \beta_i \gamma_i$, where $0 =< i =< |\alpha|$, and M_i be a marking reachable from M by firing β_i. For a formula f, $<M, \alpha> \; |==$ means that f is satisfied by the pair of M and α:

a) $<M,\alpha> \; |== p$ iff $M(p)>0$

b) $<M,\alpha> \; |== t_{fir}$ iff t is firable in M

c) $<M,\alpha> \; |== t$ iff $\alpha = \lambda$ and $t=\beta$

d) $<M,\alpha> \; |== \neg f$ iff $\neg <M,\alpha> \; |== f$

e) $<M,\alpha> \; |== f_1 \cdot f_2$ iff $<M,\alpha> \; |== f_1$ and $<M,\alpha> \; |== f_2$

f) $<M,\alpha> \; |== f_1 + f_2$ iff $<M,\alpha> \; |== f_1$ or $<M,\alpha> \; |== f_2$

g) $<M,\alpha> \; |== f_1 => f_2$ iff $<M,\alpha> \; |== f_1$ implies $<M,\alpha> \; |== f_2$

h) $<M,\alpha> \; |==$ of iff $\alpha = \lambda$ and $<M_1,\gamma_1> \; |== f$

i) $<M,\alpha> \; |== \Box f$ iff $<M_i,\gamma_i> \; |== f$ for every $0 =< i =< |\alpha|$

j) $<M,\alpha> \; |== \Diamond f$ iff $<M_i,\gamma_i> \; |== f$ for some $0 =< i =< |\alpha|$

The formula f of a temporal TPN imposes restrictions on possible firing sequences of PN. The possible firing sequences, L_R, from marking M can be defined as follows L_R (TPN, M) = $\{\alpha | \alpha \in L^{oo}(PN,M)$ and $<M,\alpha> \; |== f\}$.

The following properties of TPN, which can be easily proved, will be used in this paper:

(PR_1) $<M,\alpha> \; |== f + o\,f$ implies $<M,\alpha> \; |== \Diamond f$

(PR_2) $<M,\alpha> \; |== \Box (f_1 => \Diamond f_2) \cdot \Box (f_2 => \Diamond f_3)$ implies $<M,\alpha> \; |== \Box (f_1 => \Diamond f_3)$

(PR_3) $<M,\alpha> \; |== \Box (t_{ifir} => \Diamond t_i)$

5. Application of Reduction Techniques to Construction of Functional Abstractions

The Petri net models defined in [12] and [16] represent flows of physical resources and control information. The control information was classified into internal and export control information. The internal control information is responsible for the internal timing of the flow of physical resources along paths connecting the corresponding input and output interface places. Since this internal timing is of no concern for the description of the external functionality, the paths representing the flow of the internal control information can be removed from the discussed Petri net models. It should be noted that the internal timing would be of concern should we attempted to construct performance oriented abstractions which would retain both the external functionality and timing dependencies. After removing paths representing the internal control information, the models comprise paths of flow of the physical resources and the export control information. Paths of flow of the physical resources are terminated by the input and output interface places. Paths of flow of the export control information are terminated by transitions, which belong to the paths of flow of the physical resources, and the output interface places.

In order to minimize the number of transitions and places comprising paths of flow of the physical resources or export control information, the path structural reduction techniques [8] can be used. However, the application of these techniques should result in a reduced model retaining the external functionality as defined for the original model. Since the temporal input-output relationships are defined for the interface places, the path reduction will be based on replacing the sequence transition-place-transition by a macrotransition. In the Petri net models defined in [12] and [16], two types of the transition-place-transition sequences can be distinguished. The first group comprises sequences for which the presence of tokens on the interface places, which belong to the input bag of one transition, is directly related to the presence of tokens on the interface places which belong to the output bag of the other transition. In other words, there exists a functionally meaningful path in the environment of the model which connects these two places. This path represents the flow of a request for service and the service delivery. To the second group belong sequences for which this condition does not apply. Next, it will be shown how the path reduction technique, when applied to these two types of sequences, affects the external functionality of the model. Without losing generality, the model of figure 2 will be considered.

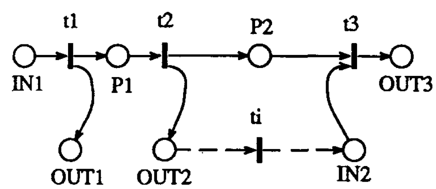

Figure 2. A simple Petri net model

Figure 2 shows a Petri net model which has two input and two output interface places. The required external functionality of this model is described as follows. If there is a token on place IN_1, then eventually there will be a token on place OUT_1 and on place OUT_2. The presence of a token on place OUT_2 will eventually result in the presence of a token on place IN_2. Finally, if there is a token on place IN_2, then eventually there will be a token on place OUT_3. The dashed line, which connects transition t_i to places OUT_2 and IN_2, indicates that t_i is possibly a part of a sequence of places and transitions. Transition t_i models the synchronization event between a request for service and the request being granted. Formally, the external functionality of the model, using the TPN notation, is given by the following formulas:

(1) $F_1 = \Box (IN_1 => \Diamond OUT_1)$
(2) $F_2 = \Box (IN_1 => \Diamond OUT_2)$
(3) $F_3 = \Box (OUT_2 => \Diamond IN_2)$
(4) $F_4 = \Box (IN_2 => \Diamond OUT_3)$

First, the path reduction is applied to the sequence comprising t_1, P_1 and t_2. Transitions t_1 and t_2 have only one interface place each. Places OUT_1 and OUT_2 belong to the output bags of t_1 and t_2, respectively. The input bags of t_1 and t_2 do not contain any input interface places. Figure 3 shows the reduced model PN_{R1}.

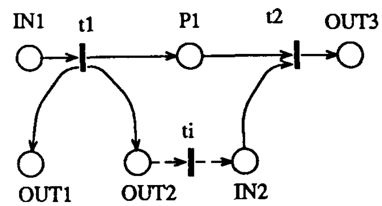

Figure 3. Reduced model PN_{R1}

Lemma 1. The reduced model, shown in figure 3, retains the external functionality of the original model.

Proof: By analysing the behaviour of the net, assuming that initially there is a token on place IN_1, we have:

(1) $\langle M,\alpha \rangle \mid== \Box (IN1 => t_{1fir})$
(1) and PR_3
(2) $\langle M,\alpha \rangle \mid== \Box (t_1 => o (P_1 \cdot OUT_1 \cdot OUT_2))$
(1), (2) and PR_1 and PR_2 yields
(3) $\langle M,\alpha \rangle \mid== \Box (IN_1 => \Diamond (P_1 \cdot OUT_1 \cdot OUT_2))$
(4) $\langle M,\alpha \rangle \mid== \Box (OUT_2 => t_{ifir})$
(4) and PR_3
(5) $\langle M,\alpha \rangle \mid== \Box (t_i => \Diamond IN_2)$
using (4), (5) and PR_2 we have
(6) $\langle M,\alpha \rangle \mid== \Box (OUT_2 => \Diamond IN_2)$
(7) $\langle M,\alpha \rangle \mid== \Box ((P_1 \cdot IN_2) => t_{2fir})$
(7) and PR_3
(8) $\langle M,\alpha \rangle \mid== \Box (t_2 => o\, OUT_3)$

(7), (8) and PR_1, PR_2 gives

(9) $<M,\alpha> \ |==\Box\ ((P_1 \cdot IN_2) => \Diamond\ OUT_3)$ []

The reduced model shown in figure 3 retains the external functional behaviour of the original model, as indicated by (3), (6) and (9).

Now, the path reduction step is applied to the sequence comprising t_2, P_2 and t_3. The output bag of t_2 contains the output interface place OUT_2. The input bag of t_3 contains the input interface place IN_2. As indicated in figure 2, the presence of a token on place IN_2 is conditioned by the presence of token on place OUT_2 ,e.g., there exists a functionally meaningful path in the environment of the model which connects these two places. Figure 4 shows the reduced model PN_{R2}.

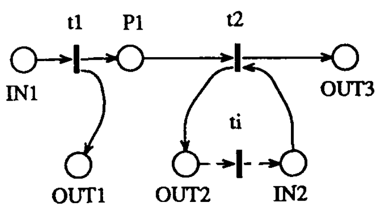

Figure 4. Reduced model PN_{R2}

Lemma 2. The reduced model, shown in figure 4, does not retain the external functionality of the original net.

Proof: The functional property which is expressed by the requirement that if there is a token on place IN_1 then eventually there will be a token on place OUT_1 can be shown to be retained, by the model of figure 4, in the same fashion as it was done in the proof of Lemma 1. More interesting, however, is the state in which there is a token on place P_1. Clearly, for t_2 to fire, there would also have to be a token on place IN_2. Hence:

(10) $<M,\alpha> \ |==\Box\ ((P_1 \cdot IN_2) => t_{3fir})$

(10) and PR_3

(11) $<M,\alpha> \ |==\Box\ (t_3 => o\,(OUT_2 \cdot OUT_3))$

(10), (11) and PR_1 and PR_2 gives

(12) $<M,\alpha> \ |==\Box\ ((P1 \cdot IN2) => \Diamond\ (OUT2 \cdot OUT3))$

(12) contradicts the requirements expressed by the formulas F_3 and F_4 of the original model. []

If OUT_2 and IN_2 represent a request for service and the requested service delivered, respectively, then (12) means that service is delivered even if it is not requested, or the completion of service is the necessary requirement for the request for service to be generated.

The above analysis shows that the reduction step can only be applied to the second type of sequences without affecting the required external functionality of the model. This observation is the basis for the formulation of an approach allowing for systematic construction of functional abstractions of the primary components of FMSs.

6. A Method for Systematic Construction of Functional Abstractions

Let PN=(P, T, IN, OUT) be a Petri net representing a model of a FMS's component and its environment; $P_M=\{\ p_{M1}, p_{M2}, ..., p_{Mk}\ \}$ be a set of places of the model; $T_M=\{\ t_{M1}, t_{M2}, ..., t_{Ml}\ \}$ be a set of transitions of the model; $P_{MI}=\{\ p_{MI1}, p_{MI2}, ..., p_{MIo}\ \}$ be a set of interface places of the model; where $P_M \cup T_M \neq \phi$, $P_M \cap T_M = \phi$, $P_M \subset P$, $T_M \subset T$, $P_{MI} \subset P_M$.

Let $P_{ME}=\{\ p_{ME1}, p_{ME2}, ..., p_{MEv}\ \}$ be a set of places of the environment of the model; $T_{ME}=\{\ t_{ME1}, t_{ME2}, ..., t_{MEw}\ \}$ be a set of transitions of the environment of the model; where $P_{ME} \cup T_{ME} \neq \phi$, $P_{ME} \cap T_{ME} = \phi$, $P_{ME} \subset P$, $T_{ME} \subset T$. Also, let $P_M \cap P_{ME} = \phi$ and $T_M \cap T_{ME} = \phi$

Definition: Transition t_{Ml} is said to be nontrivially firing-dependent on transition t_{Mi} if

$$\exists\ p_{MIi} \in OP(t_{Mi}): p_{MIi} \in IP(t_{MEk}) \text{ and}$$

$$\exists\ p_{MIl} \in IP(t_{Ml}): p_{MIl} \in OP(t_{MEk})$$

Transition t_{MEk} can possibly be replaced, by using structural refinement techniques, by a sequence of places and transitions. Two transitions are said to be nontrivially firing-dependent if there exists a functionally meaningful path in the environment of the model which connects these transitions. This path represents the flow of a request for service and the requested service delivery.

Definition: Transition t_{Mj} is said to be adjacent to transition t_{Mi} if

$$\exists\, p_{Mj} \in IP(t_{Mj}): p_{Mj} \in OP(t_{Mi})$$

Definition: Transition t_{Mj}, adjacent to transition t_{Mi}, is said to be trivially firing-dependent on transition t_{Mi} if

$$\exists\, p_{Mj} \in IP(t_{Mj}): p_{Mj} \in OP(t_{Mi}) \text{ and}$$
$$\neg \exists\, p_{MIi} \in OP(t_{Mi}): p_{MIi} \in IP(t_{MEk}) \text{ and}$$
$$\neg \exists\, p_{MIj} \in IP(t_{Mj}): p_{MIj} \in OP(t_{MEk})$$

Two adjacent transitions are said to be trivially firing-dependent if there exists no functionally meaningful path in the environment of the model which connects these transitions.

The method starts with the identification of the objects (physical resources, control information) associated with the model. Then, paths of flow of the atomic and compound objects through the model are identified. The atomic objects represent the physical resources only. The compound objects represent the export control information-physical resource (request for service-delivery of the requested service). In the next step, each path is checked for the presence of the nontrivial firing-dependences between transitions. Then, a union of these paths is constructed. Finally, for each pair of adjacent transitions, if the adjacent transitions are trivially firing-dependent then replace the two transitions and the place between them by a single macrotransition. Otherwise, if the adjacent transitions are nontrivially firing-dependent, no reduction is performed. This step is repeated until there are no trivially firing-dependent transitions left in the union. The paths of flow of the export control information-physical resource should be closed in the functional abstraction if the model composed of functional abstractions is to be analysed using structural analysis methods. The approach is shown in figure 5.

7. Systematic Construction of the Functional Abstraction of the Petri Net Model of a Machining Station

The above approach will be used in this section to obtain the functional abstraction of the Petri net model of the machining station, which can handle one part type. This model, which is shown in figure 6, was

```
BEGIN
 REPEAT
    identify the object associated with the module that
    is either a physical resource or an export control
    information-physical resource (the internal control
    information is excluded)
  REPEAT
     identify the path of flow of the object in the
     module
   REPEAT
      identify all nontrivially firing-dependent
      transitions in the path
   UNTIL all transitions
  UNTIL all paths
 UNTIL all objects

 BEGIN
 construct the union of paths
 END

  REPEAT
   IF  adjacent transitions are trivially
       firing-dependent
   THEN replace both transitions and the connecting
          place by a single transition
   ELSE no reduction ( the adjacent transitions are
          nontrivially firing-dependent)
   END
  UNTIL all adjacent transitions
END
```

Figure 5. Method for systematic construction of functional abstractions

proposed in [12]. The machining station modelled comprises a number of space resources; pallet input and output storage areas, machining area, cart unload and load areas and cart departure area. It is assumed that cart unload, part machining, cart load and cart departure areas can accommodate only one physical object. The pallet input and output storage areas can accommodate more then one pallet. In this model, a pallet present in the unload area (a token is present on place ULA) is removed from the cart (transition UL) into the input storage area (place SA-IN), provided that there is a space in this area (at least one token is present on place ISA-S). The empty cart is then

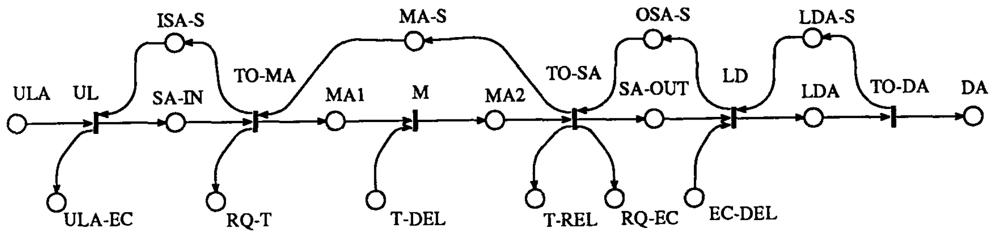

Figure 6. Petri net model of the machining station

removed from the unload area (a token is present on place ULA-EC). If the machining area is empty (a token is present on place MA-S), then a pallet is moved (transition TO-MA) to this area (a token is present on place MA1), and the request for a set of tools is generated (a token is present on place RQ-T). Once the set of tools is delivered to the machining area, the parts are machined (transition M). After machining (a token is present on place MA2), the pallet, loaded with machined parts, is moved to the output storage area (a token is present on place SA-OUT), provided that there is a space in this area (at least on token is present on place OSA-S). When the pallet is moved to the output storage area (transition TO-OSA), a request for an empty cart is generated by the machining station (a token is present on place RE-EC). Also, the set of tools is released by the machining station (a token is present on place T-REL). The delivered empty cart (a token is present on place EC-DEL) is loaded with a pallet in the load area (transition LD), provided this area is empty (a token is present on place LDA-S). After loading, the cart is moved from the load area (place LDA) to the cart departure area (place DA). This event is modelled by transition TO-DA.

In the machining station model, one can distinguish paths of flow of four objects. These objects are pallets (the movement of parts can be identified with the movement of pallets, assuming that each type of parts is allocated a dedicated group of pallets), carts, the request for a set of tools-the requested set of tools delivered and the request for an empty cart-empty cart delivered. The last two objects are compound, representing a request for service-the request granted. The path of flow of pallets is modelled by the following sequence of places and transitions ULA, UL, SA-IN, TO-MA, MA1, M, MA2, TO-SA, SA-OUT, LD, LDA, TO-DA and DA. The path of flow of carts is modelled by ULA, UL, ULA-EC and EC-DEL, LD, LDA, TO-DA, DA. The path of flow of the request for a set of tools-the requested set of tools delivered is represented by TO-MA, RQ-T and T-DEL, M, MA2, TO-SA, T-REL. The path of flow of the request for an empty cart-empty cart delivered is modelled by TO-SA, RQ-EC and EC-DEL, LD, LDA, TO-DA, DA. There are two pairs of nontrivially firing-dependent transitions in the model; TO-MA, T-DEL and RQ-EC, EC-DEL. This implies that the sequences of places and transitions TO-MA, MA1, M and TO-SA, SA-OUT, LD should not be replaced by macrotransitions in order to retain the external functionality of the original model.

The functional abstraction of the machining station model is shown in figure 7. The net representing this abstraction contains two internal places (maximum four if the paths of flow of the two compound objects are closed) and three internal transitions. The net representing the complete model of the machining station contains nine places and six transitions. The external functionality of the original model is as follows:

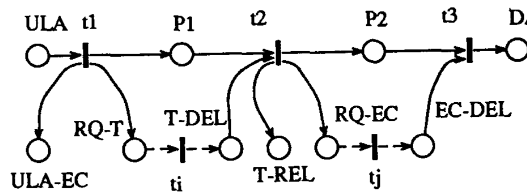

Figure 7. Functional abstractions of the machining station

(13) $MS_1 = \Box\ (ULA \Rightarrow \Diamond\ (ULA\text{-}EC \cdot RQ\text{-}T))$;

If there is a loaded cart waiting in the unload area, then eventually there will be an empty cart in the unload area and a pallet, loaded with parts, present in the pallet input storage area.

(14) $MS_2 = \Box$ (RQ-T => $\Diamond$ T-DEL)

If the request for a set of tools is generated, then eventually this set of tools will be delivered to the machining area of the station.

(15) $MS_3 = \Box$ (T-DEL => $\Diamond$ (T-REL · RQ-EC))

If the requested set of tools is delivered to the machining area, then eventually the set of tools will be released by the station and the request for an empty cart will be generated.

(16) $MS_4 = \Box$ (RQ-EC => $\Diamond$ EC-DEL)

If the request for an empty cart is generated, then eventually an empty cart will be delivered.

(17) $MS_5 = \Box$ (EC-DEL => $\Diamond$ DA)

If an empty cart is delivered to the load area, then eventually the cart, loaded with a pallet, will be present in the departure area of the station.

Lemma 3. The functional abstraction of the machining station model retains the external functionality of the original model, which is expressed by the formulas MS_1-MS_5.

Proof: The dynamic behaviour of the net, assuming that initially there is a token on place ULA, yields:

(18) $<M,\alpha> \; |== \Box$ (ULA => t_{1fir})

(18) and PR_3 yields

(19) $<M,\alpha> \; |== \Box$ (t_1 => o (ULA-EC · RQ-T · P_1))

using (18), (19) and PR_1 and PR_2 we have

(20) $<M,\alpha> \; |== \Box$(ULA => $\Diamond$ (ULA-EC · RQ-T · P1))

(21) $<M,\alpha> \; |== \Box$ (RQ-T=> $\Diamond$ t_{ifir})

(21) and PR_3 yields

(22) $<M,\alpha> \; |== \Box$ (t_i => $\Diamond$ T-DEL)

(21), (22) and PR_2 gives

(23) $<M,\alpha> \; |== \Box$ (RQ-T => $\Diamond$ T-DEL)

(24) $<M,\alpha> \; |== \Box$ ((T-DEL · P1) => t_{2fir})

(24) and PR_3 yields

(25) $<M,\alpha> \; |== \Box$ (t_2 => o (T-REL · RQ-EC · P2))

using (24), (25) and PR_1 and PR_2 we have

(26) $<M,\alpha> \; |== \Box$ ((T-DEL · P1) => $\Diamond$ (T-REL · RQ-EC · P2))

(27) $<M,\alpha> \; |== \Box$ (RQ-EC => $\Diamond$ t_{jfir})

(27) and PR_3 yields

(28) $<M,\alpha> \; |== \Box$ (t_j => $\Diamond$ EC-DEL)

(27), (28) and PR_2 gives

(29) $<M,\alpha> \; |== \Box$ (RQ-EC => $\Diamond$ EC-DEL)

(30) $<M,\alpha> \; |== \Box$ ((EC-DEL · P2) => $\Diamond t_{3fir}$)

using (30) and PR_3 we have

(31) $<M,\alpha> \; |== \Box$ (t_3 => o DA)

(30), (31) and PR_1 and PR_2 gives

(32) $<M,\alpha> \; |== \Box$ ((EC-DEL · P2) => $\Diamond$ DA) []

The external functionality, specified by the formulas MS1-MS4, of the complete model of the machining station is realized by (20), (23), (26), (29) and (32).

8. Conclusions

A method that allows one to systematically construct functional abstractions of a class of Petri net models, that represent typical components of FMSs, is presented in this paper.

The impact of functional abstractions on the verification task complexity was studied in [15]. That study shows that by replacing, in the verification model, complex subnets by simple ones, the number of places and transitions can be substantially reduced. In consequence, the number of place (transition) invariants, and places (transitions) in the corresponding invariant supports, and the size of the reachability set can be substantially reduced as well. As a result, the verification task difficulty is reduced also. Since functional abstractions represent external behaviour of modules assembled into a model, the verification effort can primarily concentrate on the model structure (interconnections of modules) and not on its components.

Although the method presented is restricted to a certain type of Petri net models, some structural and functional properties of these models seem to be common to a much wider class of Petri net models of FMSs components, and possibly other systems too. Further research into the adaptability of this method to other types of systems is still needed.

The conclusions which were drawn in section 5 point to the way in which functional abstractions of the studied class of Petri net models could be systematically synthesized from the formal description of the external functionality of these models.

Both the generic Petri net models, which were proposed in [12], and their functional abstractions were added to GreatSPN [19]. This modified Petri net tool allows for rapid modelling with generic models, verification with functional abstractions and subsequent performance evaluation of FMSs.

9. References

[1] Narahari Y., Viswanadham N., "A Petri Net Approach to the Modelling and Analysis of Flexible Manufacturing Systems", *Annals of Operations Research* vol. 3, pp 449-472, 1985.

[2] Narahari Y., Viswandham N., "A Petri Net Based Investigation of Deadlocks in Automated Manufacturing Systems", *CAD/CAM/CAE for Industrial Progress*, Ed. V.Rajaraman, Elsvier Science Publishers, North-Holland, pp. 119-131, 1986.

[3] Krogh B., Beck C., "Synthesis of Place Transition Nets for Simulation and Control of Manufacturing Systems", *Proc. IFAC Large Scale Systems: Theory and Applications*, pp. 583-588, 1986.

[4] Valette R., Courvoisier M., Demmou H., Bigou J., Desclaux C., "Putting Petri Nets to Work for Controlling Flexible Manufacturing Systems", *Proc. ISCAS 85*, pp. 929-932, 1985.

[5] Suzuki I., Murata T., "Stepwise Refinement of Transitions and Places", *Informatik-Fachberichte* 52: Application and Theory of Petri Nets, Eds. C. Giraut, W Reisig, pp. 136-141, Springer-Verlag, 1982.

[6] Suzuki I., Murata T., "A Method for Stepwise Refinement and Abstractions of Petri Nets", *J. Comp. Sys. Science*, vol. 27, no. 1, pp. 51-76, Aug. 1983.

[7] Berthelot G., "Checking Properties of Nets Using Transformations", *Advances in Petri Nets 85*, Ed. G. Rozenberg, Lecture Notes in Computer Science 222, Springer-Verlag, pp.19-40, 1986.

[8] Lee K.H., Favrel J., "Hierarchical Reduction Method for Analysis and Decomposition of Petri Nets", *IEEE Trans. on Systems, Man and Cybernetics*, vol. SCM-15, no. 2, pp. 272-281, Mar/Apr. 1985.

[9] Lee K.H., Favrel J., Baptiste P., "Generalized Petri Net Reduction Method", *IEEE Trans. on Systems, Man and Cybernetics*, vol. SCM-17, no. 2, pp. 297-303, Mar/Apr. 1987.

[10] Lee K.H., Favrel J., Oh J., "Hierarchical Decomposition of Petri Net Languages", *IEEE Trans. on Systems, Man and Cybernetics*, vol. SCM-17, no. 5, pp. 877-878, Sep/Oct. 1987.

[11] Zhou M.C., DiCesare F., Desrochers A., "A Top-down Approach to systematic synthesis of Petri Net Models for Manufacturing Systems", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 534-539, 1989.

[12] Zurawski R., Dillon T., "Specification, Verification and Performance Evaluation of Flexible Manufacturing Systems", *Computer Applications in Production and Engineering*, Eds. F. Kimura, A. Rolstadas, pp. 517-525, Elsevier Scientific Publishers, North-Holland, 1989.

[13] Baldassari M., Bruno G., "PROTOB Object-Oriented Graphical Modelling and Programming Based on PROT Nets", *Proc. 9th Inter. Conf. on Application and Theory of Petri Nets*, pp. 333-341, 1988.

[14] Hollinger D., Bel G., "An object-Oriented Approach for CIM Systems Specification and Simulation", *Computer Applications in Production and Engineering*, Ed. Bo K., pp. 445-460, Elsevier Science Publishers, North-Holland, 1987.

[15] Zurawski R., Dillon T., "Application of Functional Abstractions to Verification of Flexible Manufacturing Systems Modelled using Petri Nets", *Proc. IECON'90*, Asilomar,pp. 471-475, November 27-30, 1990.

[16] Al-Jaar R., Desrochers A., "Performance Evaluation of Automated Manufacturing Systems using Stochastic Petri Nets", *IEEE Trans. Robotics and Automation*, vol. 6, no. 6, pp. 621-639, Dec. 1990

[17] Suzuki I., Lu H., "Temporal Petri Nets and their Application to Modelling and Analysis of a Handshake Daisy Chain Arbiter", *IEEE Trans. Comp.*, vol. 38, no. 5, pp. 696-704, May 1989.

[18] Suzuki I., "Formal Analysis of the Alternating Bit Protocol by Temporal Petri Nets " *IEEE Trans. Soft. Eng.*, vol. 16, no. 11, pp.1273-1281, Nov. 1990.

[19] Chiola G., GreatSPN User's Manual, Version 1.3, Dipartimento di Informatica, Universita degli Studi di Torino, September 1987.