Parameterised Verification of Class of Resource Allocation Systems

Jabr Romhain[1] Kamel Barkaoui[2]

Abstract

The present work deals with two problems concerning behavioural properties for a large class of resource allocation systems (RAS) called G-systems generalising well-known models presented in the literature. The first problem is the well-formedness characterisation. It exists to prove the existence of an initial marking ensuring non-blockingness (from any state reachable from initial state, it is always possible to reach a desirable (or final) state). The second problem is to show that under appropriate supervision, non-blockingness of G - sytems, can also be always ensured. Using structure theory of Petri nets, we state, a structural and parameterised characterisation for these two problems. In particular, the proposed solution for the second problem can be interpreted as a synthesis of a parameterised and modular supervisor.

[1] Dep. of Electronics and Communication Engineering, Damascus University.
[2] Lab. Cedric- Cnam, Paris- France.

1. Introduction

Systems with resource sharing, where multi-process executions exist, are common in many contemporary applications such as flexible manufacturing systems, workflow management systems or computer operating systems. A challenge for researchers is to develop an optimal method ensuring the control of workflow and the resource allocation in these systems. Considerable research have been carried out on these topics. The present work can be related to the significant works adopting Petri net or graphs as formalism such [1, 4, 5, 10, and 11]. In section 2, we recall some basic notions of structure theory of Petri nets. Section 3 describes the G-systems, which are a class of resource allocation systems (RAS). Section 4 presents a necessary and sufficient parameterised condition for a G-system to be well-formed. Using a purely structural reasoning, we develop in section 4 a parameterised and modular method ensuring the non-blockingness of G-systems regardless of number of processes to be executed.

2. Basic definitions and notions of Petri Nets

In this section, we introduce the basic Petri net definitions[8] and notions used in this paper.

Definition 2.1. A *Petri net* is a tuple N = < P, T, F, W > where:

(i) $P \neq \emptyset$ is a finite set of *places* ;

(ii) $T \neq \emptyset$ is a finite set of *transitions* ;

(iii) $F \subseteq (PxT) \cup (TxP)$ is the *flow relation* ;

(iv) $W: F \rightarrow IN \wedge [W(x,y)= 0 \Leftrightarrow (x,y) \notin F]$ is the *weight function* ;

In the following, we define the marking of a Petri net.

Definition 2.2. A *marking* of a Petri net N is a function $M: P \rightarrow IN$.

The *initial marking* of N is denoted by Mo.

The pair < N, Mo> is called a *P/T system*.

Notation

$\forall x \in P \cup T, {}^{\bullet}x = \{y \in P \cup T / (y, x) \in F\}$ and $x^{\bullet} = \{y \in P \cup T / (x, y) \in F\}$

$\forall (p, t) \in P \times T: C(p, t) = W(t, p) - W(p, t)$

Definition 2.3.

A transition $t \in T$ is *enabled* in a marking M (denoted by $M\,[t\rangle$)

iff $p \in {}^{\bullet}t : M(p) \geq W(p,t)$

If transition t is enabled in marking M, it can be fired, leading to a new marking M' such that:

$\forall p \in P$: $M'(p) = M(p) + C(p, t)$. The firing is denoted by $M\,[t\rangle\, M'$

The set of all markings reachable from M is denoted by $[M\rangle$

We recall the main properties related to behaviour of Petri nets.

Definition 2.4.

Let < N, Mo> be a *P/T system*.

(i) A marking M_h is a *home state* iff $M \;\forall\; [\,M_o\rangle : M_h \in [\,M\rangle$;

(ii)<N, Mo> is *reversible* $\Leftrightarrow$ Mo is a home state ;

(iii) <N, Mo> is *bounded* $\Leftrightarrow \forall p \in P :[\exists k \in \mathbb{N}: \forall M \in [M_o\rangle, M(p) \leq k] \Leftrightarrow$ $[M_o\rangle$ is finite ;

(iv) N is struc*turally boun*ded$\Leftrightarrow$ $\forall M0$, <N, Mo> is bounded

(v) <N, Mo> is qu*asi-liv*e $\Leftrightarrow \forall t \in T : \exists M \in [\,M_o\rangle,\; M[t\rangle$;

(vi) <N, Mo> is dea*dlock-fre*e $\Leftrightarrow \forall M \in [\,M_o\rangle, \exists t \in T :\; M[t\rangle$;

(vii) <N, Mo> is *live* $\Leftrightarrow \forall t \in T : [\forall M \in [\,M_o\rangle : \exists M' \in [M\rangle,\; M'[t\rangle\,]$;

(viii) N is struc*turally live* $\Leftrightarrow \exists$ Mo, <N, Mo> is live

Definition 2.5.

A function $\nu : [M_o\rangle \rightarrow \mathbb{N}$ is a *norm* (strict) for a marking $M_h \in [M_o\rangle$ iff :

(i) $\nu(M) = 0 \;\Leftrightarrow\; M = M_h$;

(ii) $\forall M \in [M_o\rangle : [\nu(M) > 0 \;\Leftrightarrow \exists t \in T : M\,[t\rangle M' \wedge \nu(M') < \nu(M)\,]$;

We recall now basic some basic structural notions of Petri nets:

Definition 2.6.

Let N be a Petri net.

An integer vector $f \in \mathbf{Z}^{|P|}$, $f \neq 0$ is a *place invariant (p-invariant) iff it sat*isfies ${}^{t}f\,.C = 0$.

The *positive support* of f is the set of places $\|f\|^{+} = \{p \in P: f(p) > 0\}$

The *negative support* of f is the set of places $\|f\|^{-} = \{p \in P: f(p) < 0\}$

N is *conservative* $\Leftrightarrow$ $\exists$ *p-invariant* f / $||f||^{+} = P$ (N conservative $\Rightarrow$ N is structurally bounded).

Definition 2.7.

Let N be a Petri net and D be a non empty subset of places ($D \subseteq P$).

D is a *siphon* iff $D \subseteq D^{\bullet}$

D is minimal iff it contains no other siphon as a proper subset.

A place $p \in P$ is said to be *non-blocking* iff:

$p^{\bullet} \neq \emptyset \Rightarrow \text{Min}_{t \in {}^{\bullet}p}\ V(p,t)\} \geq \text{Min}_{t \in p^{\bullet}}\ \{V(p,t)\}$

Definition 2.8.

Let < N, Mo> be a P/T system and D siphon of N.

(i) D is *controlled* iff $\forall M \in [Mo\rangle$, $\exists p \in D$:$M(p) \geq \max\{W(p,t),\ t \in p^{\bullet}\}$

(ii)<N, Mo> satisfies the controlled-siphon property (*cs-property*) iff each minimal siphon of N is controlled

In order to check the cs-property, two main structural conditions (*sufficient but not necessary*) permitting to determine whether a given siphon is controlled are developed in [2]. These conditions are recalled below.

Proposition 2.1. Let <N, Mo> be a P/T system and D a siphon of N.
If one of the two following conditions holds, then D is controlled :

1. $\exists R \subseteq D$ such that : $R^{\bullet} \subseteq R^{\bullet}$, R is marked at Mo, and places of R are non-blocking

(Siphon D is said to be containing a trap R

2. $\exists$ a *P-invariant* f ($f \in \mathbf{Z}^{P}$) such that $D \subseteq ||\ f\ ||$ and $\forall p \in (||\ f\ ||^{-} \cap D)$: $V(p) = 1, ||\ f\ ||^{+} \subseteq D$, and $\sum_{p \in P} [f(p) . Mo_{(p)}] > \sum_{p \in D}[f(p). (V(p) - 1)]$

A siphon controlled by the first (second) mechanism is said to be trap-controlled (invariant controlled)

Two well-known basic (and obvious) relations between liveness properties and the cs-property are:

Proposition 2.2.

Let <N, Mo> be a P/T system. The two following properties hold :

(i) <N, Mo> is live $\Rightarrow$ <N, Mo> satisfies the cs-property.

(ii) <N, Mo> satisfies the cs-property $\Rightarrow$ <N, Mo> is deadlock-free .

Two other properties useful for behavioural analysis are:

Proposition 2.3.

Let <N, Mo> be a P/T system:

(i) M_h is homes state $\Leftrightarrow$ $\exists$ a norm for M_h.

(ii)<N,Mo>is quasi-live under Mo and Mo is a home state$\Rightarrow$<N,Mo>is live .

Definition 2.9 :

A *labelled Petri net* (or Petri net generator) is a tuple G= <N, l, Mo, M_F > where:

N= <P, T, F, W> is a Petri net structure;

l: T $\rightarrow$ Σ is a labelling function labels that assigns to each transition a label from the alphabet of events of Σ ;

Mo is an initial marking; M_F is a finite set of final markings.

The class of Resource allocation system (RAS) considered in this paper is Deterministic Discrete Event System (DES) and is represented by labelled Petri nets [6] where event set Σ can be partitioned into disjoint subsets: the set Σ_C of controllable events (events that can be prevented from happening, or disabled, by control) and the set Σ_U of uncontrollable events (events that can not be disabled by control).

3. The Resource allocation G-System

In this section, we present a class of RAS called G-systems. A **G-System** can be viewed as a labelled Petri net system describing a general problem arising in many contemporary application domains such flexible manufacturing systems or workflow management systems. It consists of a set of a finite number of shared resources types and a set P of a finite number of part types (or case types) that the system must produce (or execute) using resources.

For each part type (or case type) it is assigned a working process (or a business process) describing all the possible operation sequences for a given part-type (or a case-type). Our model can be qualified to be general since some realistic features can be described. Among them, we notice: (1) process flexibility i.e. a part-type (or a case type) can have more than one routing, (2) assembly (synchronisation) /disassembly (splitting) operations, (3) assignment flexibility i.e. a same operation can be performed with different resource-type, (4) permutation flexibility i.e. the order of a subsequent of operations is not fixed (some operations are partially ordered) -.

The Petri net based description of these concurrent processes associated to a working or a business process provides a circuit-free labelled Petri net called **G-Task** [2, 9,10 ,11]

Once internal soundness (abstraction of resource use) of each G-task is proved, we have to ensure the non blockingness of the G-System i.e. the correctness of the shared resource allocation strategies between processes belonging to same or different G -Task subsystems. The generality of the entire model enhanced by the fact that each operation may use multiple resources of different type and that major synchronisation patterns with shared resources such generalised parallel and sequential mutual exclusions are allowed. Finally, one can note that the structure of a G-system can be described by standard engineering tools as the task sequencing and resource sequencing matrices [7].

Now, we first define formally the class of DES called G-task.

Definition 3.1 : G-Task Systems (GT)

A *G-Task GT* is a labelled Petri net G = <N, l, Mo, M_F> where:

(i) N=<P, T, F, W> is a circuit-free Petri net with two special places: i and o.

Place i is a source place ($^\bullet i = \emptyset$) place o is a sink place ($o^\bullet = \emptyset$)

(ii)The augmented net **N*** obtained from N by adding a transition t* such that

$^\bullet t^* = \{o\}$ and $t^{*\bullet} = \{i\}$ (and W (o,t*) =W (t*,o)=n) is strongly connected.

(iii) Mo = n.i ; M_F = n.o

(iv) <N, n.i> is quasi-live

Definition 3.2:

A G-Task GT is sound iff :

(i) $\forall M \in [n.i\rangle$, $n.o \in [M\rangle$

(ii) $\forall M \in [n.i\rangle$, $M(o) \geq n \Rightarrow M = n.o$

Proposition 3.1:

Let GT be a G-task, GT is sound iff
<N*, n.i> is live and bounded,

Proof

$\Leftarrow$) Let us suppose <N*, n.i> is live and bounded,
$\forall M \in [n.i\rangle : \exists M' \in [M\rangle, M'(o) \geq n$. Let us suppose that $M' = n.o + M'', M'' \neq 0$.
Then, $M'[t^*\rangle\ n.i + M''$ which contradicts the boundedness hypothesis.
Therefore, M'= n.o and <N, l, n.i, n.o> is sound.

$\Rightarrow$) Let us assume <N, l, n.i, n.o> is sound. We first prove <N*, n.i> is bounded. Suppose that <N*, n.i> is not bounded. Then, $\exists M1 \in [n.i\rangle$: $\exists M2 \in [M1\rangle$, M2 > M1.

As <N, l, n.i, n.o> is sound, we know (definition 3.2 (i)) that $\exists \sigma \in T^*$: $M1[\sigma\rangle\ n.o$.

Thus, $\exists M$, $M2\ [\sigma\rangle\ M$: $M > n.o$. This contradicts the soundness hypothesis (definition 3.2 (ii)).

We now prove <N*, n.i> is live. As <N, l, n.i, n.o> is consistent, from (definition 3.2 (i)), $\forall M \in [n.i\rangle$, $n.o \in [M\rangle$. Then, by firing t*, we obtain: $\forall M \in [n.i\rangle$: $n.i \in [M\rangle$, i.e. n.i is a home state of <N*, n.i>. As by definition 3.1, <N*, n.i> is quasi-live, then, by proposition 2.3 (ii), <N*, n.i> is live.

Definition 3.3:

A G-Task GT is well-formed iff:
$\exists$ Mo = n.i such that GT is sound

Theorem 3.1:

Let GT be a G-task, GT is well-formed iff
$\exists$ Mo = n.i such that <N*, n.i> is bounded and satisfies the cs-property.

Proof

$\Rightarrow$) Let us suppose that G is well-formed. From proposition 3.1., <N*, n.i> is live and bounded, in particular, it satisfies the cs-property (proposition 2.1).

$\Leftarrow$) Let us suppose that $\exists$n : <N*, n.i> is bounded and satisfies the cs-property.

We first exhibit a norm ν for marking n.o. We construct function ν as follows : we number the places of the net N in reverse topological order, i.e. place o is numbered 0, place i has the highest number, and the other places are such that a successor p' of a place p in the graph of the Petri net has a lower number than place p. This can be done since N is circuit free. We call Num this numbering function. Then, we define $\forall$M : $\nu(M) = \Sigma_{p\in P}$ M(p).Num(p).

We now prove that ν is a norm for n.o. By construction, $\nu(M) = 0 \Leftrightarrow$ M = x.o. If x < n, then there is no t $\in$ T : M[t$\rangle$. As <N*, n.i> satisfies the cs-property, we deduce from proposition 2.1.(ii) that <N*, n.i> is deadlock-free, i.e.$\forall$M $\in$ [n.i$\rangle$, $\exists$t $\in$ T : M[t$\rangle$. Thus, there is a contradiction. If x > n, x.o[t*$\rangle$M' > n.i. This contradicts the boundedness hypothesis. Thus, we have proved condition (i) of definition 2.5.

Let us suppose that ν(M) > 0 for a marking M. As (N*n n.i> is deadlock-free, $\exists$t. $\in$ T : M[t$\rangle$M'. If t $\neq$ t*, by construction of ν, ν(M') < ν(M). Otherwise (t = t*), as ν(M) >0 and ν(M) = 0 $\Leftrightarrow$ M = n.o (already proven), marking M must have the form M = n.o + M'' with M'' $\neq$ 0. Then M[t*$\rangle$n.i + M'' >n.i, which contradicts the boundedness hypothesis. Thus, $\Rightarrow$ of definition 2.5.(ii) is satisfied.

Let us now suppose that $\exists$t $\in$ T: M[t$\rangle$M' $\wedge$ ν(M') < ν(M). The construction of function ν is such that $\forall$M : ν(M) $\geq$ 0. Then ν(M) > ν(M') $\geq$ 0. Thus, $\Leftarrow$ of definition 2.5. (ii) is satisfied.

We deduce from all that function ν is well a norm for n.o. From proposition 2.3. (i), n.o is a home marking. As <N*, n.i> is quasi-live and its initial state is a home state, it is live. From proposition 3.1., <N, n.i> is well-formed.

Now, we define the class of G-Task with resources (GTR systems). A GTR systems is basically a consistent G-Task plus a set of places (P_R) modelling the resources shared by its processes.

We demand the G-task net with resources to be (externally) sound with respect to resource use i.e. a resource requested will eventually be released and a resource released has previously been requested. This resource preservation property is expressed in terms of invariants in the system (definition 3.4.(v)). Due to structure of G-Task subsystems, subnets induced by these invariants are not necessarily state machines. We recall that several resources can be requested/ released simultaneously.

Definition 3.4: G-Task systems with Resources (GTR)
A G-Task system with Resources GTR is a labelled Petri net
GR= $<NR, l, MR_o, MR_F>$ where:

(i) NR= $< P \cup P_R, T, F \cup F_R, W \cup W_R>$ (the associated Petri net structure)

(ii)$P_R \neq \emptyset$ and $P \cap P_R = \emptyset$ (P_R is the set of resources)

(iii) $F_R \subseteq (P_R \text{ x } T) \cup (T \text{ x } P_R)$ (the flow relation for resources)

(iv) $\forall u \in F_R, W_R(u) \geq 1$ (resource use)

(v) $\forall r \in P_R, \exists f_r = 0$: ${}^t f_{r.} C = 0$ and $\|f_r\| \cap P_R = \{r\}$ (resource preservation)

(vi) $MR_o= n.i + \Sigma k_j.r_j$, $MR_F= n.o + \Sigma k_j.r_j$ (o, i ΣP ; r_j P_R, j=1,$|P_R|$)

(vii) The subsystem G=< N, Mo, M_F, l>, where N=<P, T, F, W> and Mo and M_F are respectively the restrictions of MR_o and MR_F to P, is a consistent G-task.

Then, we can compose several GTR nets into a system where they share resources. This is obtained by fusion of the places representing the resources shared by different GTR systems.

Definition 3.5: G-Systems (GS)
a G-System GS is recursively defined. A GTR is a G-System
Let GS_i = $<NS_i, l_i, MS_{oi}, MS_{Fi}>$, $i \in \{1, 2\}$, be two G-Systems such that $P1 \cap P2 = T1 \cap T2 = \emptyset$. We denote the set of shared resources by $P_{R1}P_{R2}= P_{R1} \cap P_{R2}$.
The system GS = GS_1 o GS_2 resulting of the fusion of systems GS_1 and GS2 over the set $P_{R1}P_{R2}$ is a G-System

A relevant property of any system with resource sharing is to be non-blocking i.e. from any state reachable from initial state; it is always possible to reach a desirable (or final) state. In the following section, we first prove that a G-system is well-formed, i.e. there exits an initial marking for which it is non-blocking.

4. A Parameterised characterisation of well-formed G-Systems

a G-system GS is *well-formed* iff there exists an initial marking MS_O such that :

$\forall M \in [MS_O \rangle , MS_F \in [M\rangle$

We prove below that a given G-system is well-formed.

Proposition 4.1. Let D a minimal siphon of NS*. There exists an initial marking M_O under which D is controlled.

Proof: Let GS be a G-system and D a minimal siphon of NS*. The augmented net NS* is obtained from NS by augmenting the net N_i of each G-Task (component of GS) as defined in 3.1 (ii)..Let us first suppose that $D \cap \cup P_{Ri} = \emptyset$. By construction and due to minimality of D there exists a G-Task subsystem GT_i such that $D \subseteq P_i$ and D contains the input place of GT_i . As GT_i is assumed to be sound, there exists an initial marking under which D is controlled. Let us consider now the complementary case $D \cap \cup P_{Ri} \neq \emptyset$ and suppose now that siphon D is not controlled. We denote by f(r) a flow of minimal support associated with resource r, and by f(p) a flow of minimal support associated with p in its corresponding G-task net.

Let : $g_D = \Sigma f(r)$, $r \in D \cap \cup P_{Ri}$; $Out(D)= \|g_D\| \setminus D$; $h_D = \Sigma f(p)$, $p \in Out(D)$;

$\lambda_D =$ max g(p) , $p \in Out(D) \cap \|h_D\|$; $z_D = g_D - \lambda_D . h_D$;

Siphon D is controlled as soon as ${}^t z_D . M_O > \Sigma z_D(p) . (\max_{p^{\bullet}}\text{-}) , p \in D$

Therefore, there exists a marking under which D is controlled.

Theorem 4.1.

Let GS be a G-system. GS is well-formed iff there exists an initial marking M_O under which $<NS^*, M_O>$ is bounded and satisfies the cs-property (a such marking is called a controlled marking).

Proof:

$\Rightarrow$) obvious

$\Leftarrow$) Let GS be a G-system. Let us suppose that there exists an initial marking M'o under which <NS*, M'o > is bounded and satisfies the cs-property. We prove now that which <NS*, M'o > is live. To do that, we proceed as in the proof of theorem 3.1., i.e. we exhibit a norm ν_R.

This norm ν_R is an extension of norm ν where the resources are numbered 0. Hence, GS is well-formed.

4.2. Example

We will now apply the previous theorem to the G-System (it is also a GTR) of figure 1.

First, we check the soundness of the associated G-Task: the corresponding augmented net N* contains the two following minimal siphons D1= {i, p1, p3, p5, p6, o} and D2= {i, p2, p3, p4, p5, p6, o}.

For M_0 (i) > 0, these two siphons are trap controlled. Then the associated G-task is well sound.

Now, consider the four minimal siphons of the *augmented net* NS*:

D3= {p3, p5, p6, r1}; D4= {p4, p5, p6, r2}; D5= {r1, r2, p5, p6}; D6= {i, p1, p5, p6, o, r2}.

Siphons D3 and D4 are the support of positive flow, they are invariant controlled by definition.

Their controllability conditions are M_0 (r1) > 1 for D3 and M_0 (r2) > 0 for D4.

We now have to determine a controllability condition for siphons D5 and D6 as indicated in proposition 4.1.

Consider first siphon D5= {r1, r2, p5, p6}:

g_{D5} = f(r1) + f(r2) = r1 + 2.p3 + 2.p5 + p6 + r2 +p4 + p5 + p6 = r1 + r2 + 2.p3 + p4 + 3.p5 + 2.p6

Out (D5) = {p3, p4};

h_{D5} = f (p3) + f (p4) = i + p1 + p3 + p5 + p6 + o + i + p2 + p3 + p4 + p5 + p6 + o

= 2.i + p1 + p2 + 2.p3 + p4 + 2.p5 + 2.p6 + 2.o; λ_{D5} = 2; λ_{D5} = 2;

z_{D5} = r1 + r2 + 2.p3 + p4 + 3.p5 + 2.p6 - 4.i -2.p1 - 2.p2 - 4.p3 - 2.p4 - 4.p5 - 4.p6 - 4.o

= r1 + r2 - 4.i -2.p1 - 2.p2 - 2.p3 - p4 - p5 - 2.p6 - 4.o

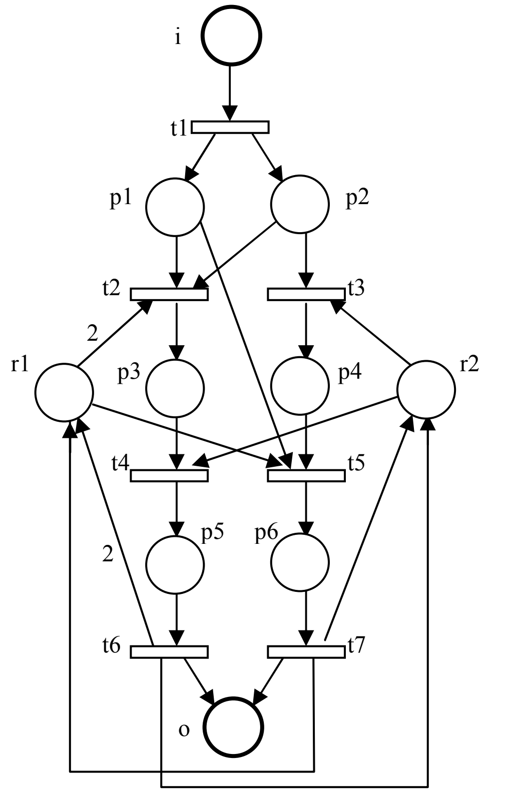

Figure1. Example of a G-system

Hence, for M_0 (r1) + M_0 (r2) - 4.M_0 (i) > 1, siphon D5 is controlled. In the same manner, we obtain a control condition for D6= {i, p1, p5, p6, o, r2}: M_0 (r2) - M_0 (i) > 0.

Finally, we can conclude that for every initial marking satisfying the following initial conditions : M_0(r1)>0 ; M_0(r2)>0 ; M_0(i) > 0 ; M_0(r1) + M_0(r2) - 4.M_0(i) > 1 and M_0(r2) - M_0(i) > 0
this G-System of figure1 is well-formed.

5. A Parameterised non-blockingness characterisation of G-Systems

In the previous section, we have proved that G-systems are well-formed; however the existence of controlled markings depends of the initial marking of inputs places of G-Task subsystems (number of parts or cases of each type to be executed). In practice, these input places must be viewed as being a part of environment, so we have to ensure non-blockingness in a manner which is independent from markings of these input places. We solve this control problem in this section, by a method based on a purely structural reasoning. This method can be interpreted as a parameterised and modular synthesis of a supervisor ensuring the non-blockingness independently of initial markings of input places.

The set of minimal siphons of such G-system can be partitioned into three classes

The first class (type 1) contains the minimal siphon without resources places. They are controlled since the G-task nets constituting the G-system are non-blocking.

The second class (type 2) contains those which include resources and are invariant controlled by construction.

The last class (type 3) contains the minimal siphons including resource places but not necessarily invariant controlled for the initial marking. Hence, due to the structure of G-systems, only the non-controllability of minimal siphons of type 3 are the rudimentary causes of blockingness.

We associate with each siphon D of type 3, a local control place C_D with:

$C_D^\bullet = {}^\bullet Out\ (D)$, ${}^\bullet C_D = Out\ (D)^\bullet$ $\forall p \in Out\ (D), \forall t \in {}^\bullet p, \forall t' \in p^\bullet$: $W\ (C_D, t) = W\ (t', C_D) = g\ (p)$

One can easily avoid self loops introduced by the flow relation restricted to C_D, since this operation preserves the invariant and thus the future control. Adding place C_D has created a new flow:

$f(C_D) = C_D + \Sigma\ g(p).p$, $p \in Out(D)$. Let $z_{C_D} = g_D - f(C_D)$.

For siphon D to be controlled, we must have: ${}^t Z_{c_D} .M_0 > \Sigma\ Z_{c_D}\ (p).(max_{p^\bullet}\ -1)$.

It is important to note that these new control places behave like resources, i.e. they satisfy the resource preservation condition of definition 3.4(v). Hence, the initial G-system augmented with these new control places remains a G-system. The subnet (not necessarily connected) induced by added control places can be viewed as a modular supervisor S. The flow relation of control places is such that arcs from a control place to uncontrollable transitions (Σ_U) are not allowed. Such construction can be algebraically determined: Indeed, in the case of presence of uncontrollable transitions in $^\bullet$Out (D), the cardinality of the support of the minimal flow f can be always minimally enlarged for satisfying the condition $C_D{}^\bullet \subseteq \Sigma_C$.

Now, one could object that these new control places can create siphons of type 3, i.e. minimal siphons including resource places but not invariant controlled. If this is the case, we have to control them as it is previously done. We can ensure that this iterative control process necessarily stops:

Indeed, for each new minimal siphon of type 3, it is associated a flow f with $\| f \| \setminus CD \subseteq \cup Pi$,

As the power set of $\cup$Pi is finite *(the dimension of the generator family of flow with minimal support is finite),* we reach necessarily a step where the role of the control place C' to be added can be played by an already existing control place C for which marking must be updated (i.e. a supervisor is constructed): [$C^\bullet = C'^\bullet$ and $^\bullet C = {}^\bullet C'$. Mo(C) = Min (Mo (C), Mo (C'))].

Hence, the non-blockingness of the class of G-Systems, even with the presence of uncontrollable transitions, can be done using a purely structural reasoning.

6. Conclusions

In this work, we presented a model describing a large class of resource allocation systems namely G-systems, and we presented how one can take advantage from recent results of structure theory of Petri nets to cope with blocking problems due to the existence of general use of shared resources. The proposed approach has advantages compared to approaches based on the computation of reachability set. Indeed, the solutions obtained by our method, are modular and parameterised. In case of internal marking modification (availability of resources, or scalability of the system), the verification of non-blockingness conditions requires only to compute again the initial markings of control places. Finally, the complexity of the presented synthesis method is reducible to the complexity of the algorithm for computing minimal siphons in Petri nets. One of our future research topics is to develop an efficient algorithm to compute minimal siphons of a G-system by exploiting its structure.

References

[1] K. Barkaoui, A. Chaoui and B. Zouari, "Supervisory Control of Discrete Event Systems based on Structure Theory of Petri Nets", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics SMC, 1997.

[2] K. Barkaoui and J.F.Peyre,"On Liveness and Controlled Siphon Property", Proceedings of the 17th International Conference on Application and Theory of Petri Nets, LNCS n° 1091, Springer 1996.

[3] C. G. Cassandras and S.Lafortune, "Introduction to Discrete Event Systems", Kluwer Academic Publishers, 1999.

[4] J. Ezpeleta, F. García-Vallés and J.M.Colom,"A Class of Well Structured Petri Nets for Flexible Manufacturing Systems", Proceedings of the 19th International Conference on Application and Theory of Petri Nets, LNCS Vol n°.1420, Springer, 1998.

[5] M.P Fanti, B. Maione, S. Mascolo, B. Turchiano, "Event-Based Feed-back Control for Deadlock Avoidance in Flexible Manufacturing Systems" IEEE Trans. On Robotics and Auto, Vol. 13, pp 347-363,1997.

[6] A.Giua and F. DiCesare, "Blocking and Controllability of Petri nets in supervisory control", IEEE Transactions on automatic control 39, N°4, pp 818-823, 1994.

[7] A. Kusiak, "Intelligent Scheduling of Automated machining Systems" in Intelligent Design and Manufacturing, Ed. Wiley, 1992.

[8] P.J. Ramadge and W.M Wonham,"The Control of discrete event systems, Proceeding of IEEE, Volume.77, n°1, pp 81-98, 1989.

[9] W. Reisig, "Petri Nets, an Introduction.EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1985.

[10] S.A Reveliotis and P.M Ferreira," Deadlock Avoidance policies for automated Manufacturing cells", IEEE Trans. On robotics and Automation, Vol. 12, No. 7, 1996.

[11] M.C Zhou and F. DiCesare, "Petri net synthesis for Discrete Event Control of Manufacturing Systems", Kluwer academic Publishers, 1993.

[12] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, F. B. Vernadat, "Practice of Petri in Manufacturing", Chapman & Hall, 1993.

Received, 10 Julie, 2004.