Structural Analysis of Resource Allocation Systems with Synchronization Constraints

Spyros Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
USA

Abstract – **The work presented in this paper extends recently developed results in the theory of resource allocation systems (RAS) to RAS behaviors that present synchronization requirements in the underlying process flows. More specifically, the paper first provides a formal definition of the considered RAS structures and behaviors by introducing the Petri net sub-class of Generalized Augmented Marked Graphs (G-AMG), and subsequently, it proceeds to the analysis of the liveness properties of the resource allocation underlying the G-AMG operation. It is shown that similar to the case of some previously studied RAS structures, non-liveness of the resource allocation taking place in the G-AMG context can be attributed to a structural object known as resource-induced deadly marked siphon, that is developed in a properly defined projection of the net reachability space. Beyond its theoretical significance in terms of formally characterizing and understanding the nature of deadlock or livelock that can occur in the considered resource allocation context, the aforementioned result is also deemed to be important from the practical standpoint of synthesizing liveness-enforcing supervisors (LES), since it is expected to enable LES synthesis through application of the correctness verification methodology originally developed in [5]; the actual potential of this approach is currently under investigation.**

I. INTRODUCTION

Motivated by the ever increasing significance of the problem of characterizing and enforcing the liveness of sequential resource allocation systems (RAS) that underly many contemporary technological applications [1], the work presented in this paper seeks to extend the existing results on the Petri net (PN) based structural characterization of RAS liveness (e.g., [2], [3], [4], [5]), to RAS that involve synchronization constraints, similar to those arising in assembly and/or disassembly operations. Furthermore, similar to the work presented in [5], it is assumed that the resource allocation requests associated with the processing stages of the various (sub-)processes are structured quite arbitrarily in terms of the number of the involved resource types and units. The resulting RAS class will be characterized as the *Assembly/Disassembly (A/D-)* RAS.

Our key finding is that, similar to the case of the Conjunctive/Disjunctive-RAS[1] studied in [5], the lack of liveness of the resource allocation taking place in the A/D-RAS can be explained in the Petri net modeling framework by the development of a particular structural object known as *resource-induced deadly marked siphon*, which is detected, however, in a modified reachability space that constitutes a certain projection of the original net reachability space. Beyond its theoretical significance in terms of formally characterizing and understanding the nature of deadlock or livelock that can occur in the considered resource allocation context, this finding is also important from the practical standpoint of synthesizing liveness-enforcing supervisors (LES), since it is expected to enable LES synthesis through application of the correctness verification methodology originally developed in [5], an issue currently under investigation. Finally, we notice that some preliminary results regarding the structural characterization of liveness in the A/D-RAS context have been reported in [3], [4]; however, those works constrain the resource allocation taking place at each process stage to a single resource unit at time, and therefore, the developed results are of more limited applicability.[2]

II. ASSEMBLY/DISASSEMBLY RAS AND GENERALIZED AUGMENTED MARKED GRAPHS

This section first revises the Petri net (PN) related concepts that are necessary for the formal modeling and analysis of the A/D-RAS, and subsequently, it provides a detailed PN-based characterization of these resource allocation environments introducing the PN class of Generalized Augmented Marked Graphs. For a more extensive discussion on the PN-based modeling framework and the currently available tools for structural and behavioral analysis of the resulting models, the reader is referred to [7], [8].

---

[1] The *Conjunctive/Disjunctive (C/D)-RAS* models sequential RAS behavior that allows (i) the association of arbitrarily structured resource allocation requests with the various processing stages, and (ii) routing flexibility. It is studied extensively in [5], and currently it is one of the broadest RAS classes to be systematically studied in the literature.

[2] In fact, they can be considered as the specialization of the results presented herein to the more constrained A/D-RAS sub-class studied in those papers; c.f. [6] for a more technical characterization of this relationship.

A. Petri net Preliminaries

A marked Petri net (PN) is defined by a quadruple $N = (P,T,W,M_0)$, where P is the set of *places*, T is the set of *transitions*, $W: (PxT) \cup (TxP) \rightarrow \{0,1,2,\ldots\} \equiv Z^+$ is the *flow relation*, and $M_0: P \rightarrow Z^+$ is the net *initial marking*, assigning to each place $p \in P$, $M_0(p)$ *tokens*. In the special case that the flow relation W maps onto {0,1}, the Petri net is said to be *ordinary*. The set of input (resp., output) transitions of a place p is denoted by ${}^\bullet p$ (resp., $p^\bullet$). Similarly, the set of input (resp., output) places of a transition t is denoted by ${}^\bullet t$ (resp., $t^\bullet$). This notation is also generalized to any set of places or transitions, X, e.g., ${}^\bullet X = \cup_{x \in X} {}^\bullet x$. The ordered set $X=<x_1, x_2,\ldots,x_n> \subseteq P \cup T$ is a path, if and only if (iff) $x_{i+1} \in x_i^\bullet$, i=1,…,n-1. Furthermore, a path is characterized as a circuit or cycle iff $x_1 \equiv x_n$. Finally, an ordinary PN such that (s.t.) $\forall t \in T$, $|t^\bullet|=|{}^\bullet t|=1$ (resp., $\forall p \in P$, $|p^\bullet|=|{}^\bullet p|=1$), is characterized as a *state machine* (resp., *marked graph*).

Given a marking M, a transition t is *enabled* iff $\forall p \in {}^\bullet t$, $M(p) \geq W(p,t)$, and this is denoted by M[t>. A transition $t \in T$ is said to be *disabled* by a place $p \in {}^\bullet t$ at M iff $M(p) < W(p,t)$. Furthermore, a place $p \in P$ for which $\exists t \in p^\bullet$ s.t. $M(p) < W(p,t)$ is said to be a *disabling* place at M. Firing an enabled transition t results in a new marking M', which is obtained by removing W(p,t) tokens from each place $p \in {}^\bullet t$, and placing W(t,p') tokens in each place $p' \in t^\bullet$. The set of markings reachable from M_0 through any fireable sequence of transitions is denoted by $R(N,M_0)$. A marked PN N with initial marking M_0 is said to be *bounded* iff all markings $M \in R(N,M_0)$ are uniformly bounded by some constant K, while N is said to be *structurally bounded* iff it is bounded for any initial marking M_0. N is said to be *reversible* iff $\forall M \in R(N,M_0)$, $M_0 \in R(N,M)$.

In case that a marked PN is *pure* – i.e., $\forall (x,y) \in (PxT) \cup (TxP)$, $W(x,y) > 0 \Rightarrow W(y,x) = 0$ – the flow relation can be represented by the *flow matrix* $\Theta = \Theta^+ - \Theta^-$, where $\Theta^+[p,t]=W(t,p)$ and $\Theta^-[p,t]=W(p,t)$. A *p-semiflow* u is a |P|-dimensional vector satisfying $u^T \cdot \Theta = 0$ and $u \geq 0$, and a t-semiflow v is a |T|-dimensional vector satisfying $\Theta \cdot v = 0$ and $v \geq 0$. A p-semiflow u (resp., t-semiflow v) is said to be minimal iff there is no p-semiflow u' (resp., t-semiflow v') such that $\|u'\| \subset \|u\|$ (resp., $\|v'\| \subset \|v\|$), where $\|u\| = \{p \in P: u(p)>0\}$ (resp., where $\|v\| = \{t \in t: v(t)>0\}$).

Given a marked PN $N=(P,T,W,M_0)$, a transition $t \in T$ is *live* iff $\forall M \in R(N,M_0)$, $\exists M' \in R(N,M)$ s.t. M'[t>, and $t \in T$ is *dead* at $M \in R(N,M_0)$ iff there is no $M' \in R(N,M)$ s.t. M'[t>. A marking $M \in R(N,M_0)$ is a (total) *deadlock* iff $\forall t \in T$, t is dead. A marked PN is *quasi-live* iff $\forall t \in T$, $\exists M \in R(N,M_0)$ s.t. M[t>, it is *weakly live* iff $\forall M \in R(N,M_0)$, $\exists t \in T$ s.t. M[t>, and it is *live* iff $\forall t \in T$, t is live. Of particular interest for the liveness analysis of marked PN is a structural element known as *siphon*, which is a set of places $S \subseteq P$ s.t. ${}^\bullet S \subseteq S^\bullet$. A siphon is *minimal* if there is no other siphon S' s.t. $S' \subset S$. A siphon is said to be *empty* at marking M iff $M(S) \equiv \Sigma_{p \in S} M(p) = 0$, and it is said to be *deadly marked* at marking M, iff $\forall t \in {}^\bullet S$, t is disabled by some $p \in S$ [5]. Obviously, empty siphons are deadly marked siphons. Furthermore, if S is a deadly marked siphon at some marking M, then (i) $\forall t \in {}^\bullet S$, t is a dead transition in M, and (ii) $\forall M' \in R(N,M)$, S is deadly marked. Finally, it is shown in [5] that if marking $M \in R(N,M_0)$ is a total deadlock, then the set S of disabling places in M constitutes a deadly marked siphon (generalizing, thus, the relationship between total deadlocks and empty siphons in ordinary PN's).

B. Generalized Augmented Marked Graphs

For the purposes of this work, the A/D-RAS is modeled by a PN sub-class to be called *Generalized Augmented Marked Graph (G-AMG)*. Here, first we provide a formal characterization of the G-AMG sub-class, and subsequently we indicate how this characterization translates to a set of modeling assumptions regarding the operation of the A/D-RAS.

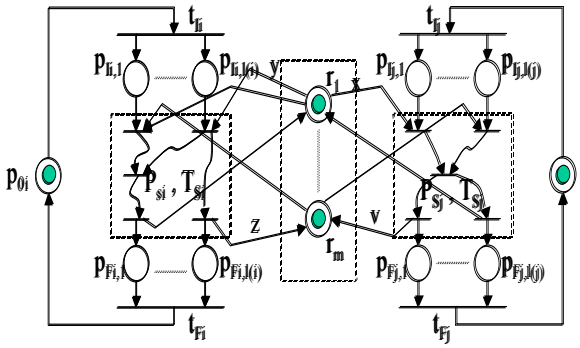

Fig. 1: The G-AMG net structure

Definition 1: A *Generalized Augmented Marked Graph (G-AMG)* is a marked PN $N=(P,T,W,M_0)$ such that

1. $P = P_S \cup P_I \cup P_F \cup P_0 \cup P_R$, where $P_X = \cup_{i=1}^n P_{Xi}$, $\forall X \in \{S,I,F\}$, $P_0 = \cup_{i=1}^n \{p_{0i}\}$, $P_R = \{r_1,\ldots,r_m\}$, and $P_X \cap P_Y = \varnothing$, $\forall X,Y \in \{S_i: i=1,\ldots,n\} \cup \{I_i: i=1,\ldots,n\} \cup \{F_i: i=1,\ldots,n\} \cup \{0,R\}$ with $X \neq Y$.
2. $T=T_S \cup T_I \cup T_F$, where $T_S = \cup_{i=1}^n T_{Si}$, $T_I = \cup_{i=1}^n \{t_{Ii}\}$, $T_F = \cup_{i=1}^n \{t_{Fi}\}$, and $T_X \cap T_Y = \varnothing$, $\forall X,Y \in \{S_i: i=1,\ldots,n\} \cup \{I,F\}$ with $X \neq Y$.
3. $W: (PxT) \cup (TxP) \rightarrow Z^+$ satisfies the following requirements:
   a. $((P_S \cup P_I \cup P_F \cup P_0)xT) \cup (Tx(P_S \cup P_I \cup P_F \cup P_0)) \rightarrow \{0,1\}$ s.t. $\forall j \neq i$, $((P_{Sj} \cup P_{Ij} \cup P_{Fj} \cup P_{0j})xT_i) \cup (T_i x(P_{Sj} \cup P_{Ij} \cup P_{Fj} \cup P_{0j})) \rightarrow \{0\}$.
   b. $\forall i$, $(\{p_{0i}\}x(T_{Si} \cup \{t_{Ii},t_{Fi}\})) \rightarrow \{1\}$ for (p_{0i},t_{Ii}), and $\{0\}$ otherwise. Similarly, $\forall i$, $((T_{Si} \cup \{t_{Ii},t_{Fi}\})x\{p_{0i}\}) \rightarrow \{1\}$ for (t_{Fi},p_{0i}), and $\{0\}$ otherwise.
   c. $\forall i$, $(\{t_{Ii}\}xP_{Ii}) \rightarrow \{1\}$ and $(\{t_{Ii}\}x(P_{Si} \cup P_{Fi})) \rightarrow \{0\}$. Similarly, $\forall i$, $(P_{Fi}x\{t_{Fi}\}) \rightarrow \{1\}$ and $((P_{Si} \cup P_{Ii})x\{t_{Fi}\}) \rightarrow \{0\}$.
   d. $(P_R x T_S) \cup (T_S x P_R) \rightarrow Z^+$ and $(P_R \times (T_I \cup T_F)) \cup ((T_I \cup T_F)xP_R) \rightarrow \{0\}$.
4. $\forall i$, i=1,…,n, the subnet N_i generated by $P_{Si} \cup P_{Ii} \cup P_{Fi} \cup \{p_{0i}\} \cup T_{Si} \cup \{t_{Ii}, t_{Fi}\}$ is a strongly connected marked graph s.t. every cycle contains $\{p_{0i}\}$.

5. $\forall r \in P_R$, $\exists$ a minimal integral p-semiflow u_r s.t. $||u_r|| \cap P_R = \{r\}$, $||u_r|| \cap (P_0 \cup P_I \cup P_F) = \varnothing$, $||u_r|| \cap P_S \neq \varnothing$, and $u_r(r)=1$.
6. N is pure and strongly connected.
7. M_0: $P \rightarrow Z^+$ with $M_0(p) \geq 1$, $\forall p \in P_0 \cup P_R$, and $M_0(p)=0$, otherwise.

♦

The basic structure implied by the above definition of G-AMG nets is depicted in Figure 1. The entire net is decomposed to n *process subnets*, interconnected through the set of resource places P_R. The elements of P_R are in one-to-one correspondence with the various *resource types* available in the underlying A/D-RAS, and their marking during the evolution of the net dynamics models the *availability* of these resources in the original A/D-RAS. In particular, $\forall k$, $M_0(r_k) \equiv C_k$, i.e., the initial marking of place r_k defines the *capacity* of the corresponding resource type. On the other hand, each process subnet N_i is further decomposed to its *idle place* p_{0i} and a remaining *acyclic marked graph* (c.f. item 4 in Definition 1) that expresses the sequential logic of the corresponding job type. Place p_{0i} contains all the tokens corresponding to process instances waiting to be loaded into the system, and it is also the final destination of all the process instances that have completed their execution into the system. Its introduction to the process subnet is a typical convention adopted in all the relevant PN-based RAS models, allowing the explicit representation of all the process instances that are re-circulated in the considered process subnet, defining thus one of the system invariants (c.f., for instance, [2], [3], [5]). The initiation of the execution of a process instance situated in place p_{0i} is represented by the firing of transition t_{Ii}, which deposits one token in each place $p_{Ii,k} \in P_{Ii}$, and initiates, thus, an instance from each sub-process involved in the execution of process net N_i. The entire sequential logic of process net N_i is defined by the subnet generated by $P_{Si} \cup T_{Si}$, while the involved resource allocation is expressed by the connectivity of the transitions in T_{Si} to the places in P_R. The execution of any process instance is considered to be complete only when all the required outcomes have been obtained. This completion requirement is expressed by the presence of a token in each place $p_{Fi,k} \in P_{Fi}$; the completion event itself is expressed by the firing of transition t_{Fi}, which returns a single token to place p_{0i}. Finally, the next definition extends to the class of G-AMG, the notion of modified marking, originally introduced in [5] for analyzing the liveness of the PN subclass modeling the behavior of C/D-RAS.

Definition 2: Given a G-AMG net $N=(P_S \cup P_I \cup P_F \cup P_0 \cup P_R, T_S \cup T_I \cup T_F, W, M_0)$, the *modified* marking $\underline{M}(p)$ is defined by $\underline{M}(p)=M(p)$, if $p \notin P_0$, and 0, otherwise. Furthermore, the set of all modified markings induced by the net reachable markings in defined by $\underline{R(N,M_0)} = \{\underline{M}: M \in R(N,M_0)\}$. ♦

III. STRUCTURAL ANALYSIS AND CONTROL OF G-AMG NETS

This section first investigates the liveness, quasi-liveness and reversibility of G-AMG nets, and their relationship to the structural concept of deadly marked siphon. At the same time, the presented analysis systematizes and generalizes the methodology developed in [5] for deriving similar structural characterizations of liveness and reversibility in the class of PN's modeling C/D-RAS. The last part of the section discusses how the derived structural characterization of liveness can support the synthesis of *liveness-enforcing supervisors (LES)* for the considered RAS class.

A. Siphon-based Characterization of Liveness in G-AMG nets

We start the discussion on the siphon-based characterization of liveness in G-AMG nets with the following lemma, which is instrumental for the development of all the subsequent results.

Lemma 1: Consider a G-AMG net $N=(P_S \cup P_I \cup P_F \cup P_0 \cup P_R, T_S \cup T_I \cup T_F, W, M_0)$. If $\exists M \in R(N,M_0)$ s.t. $M \neq M_0$ and the corresponding modified marking $\underline{M}$ is a total deadlock, then $\exists$ siphon S s.t.

i. S is deadly marked at $\underline{M}$;
ii. $S \cap P_R \neq \varnothing$;
iii. $\forall p \in S \cap P_R$, p is a disabling place at $\underline{M}$.

Proof: Let S denote the set of disabling places in modified marking $\underline{M}$. Since $\underline{M}$ is a total deadlock, S is a deadly marked siphon [5].

To establish that $S \cap P_R \neq \varnothing$, first notice that items (4) and (7) in Definition 1 imply that each process subnet N_i is a live marked graph (c.f. [8], Theorem 3.15). Furthermore, the special structure of the circuits in N_i, i=1,…,n, presumed by item 4 of Definition 1, combined with the facts that $M \in R(N,M_0)$ and $M \neq M_0$, imply that any removal of tokens from places p_{0i}, I=1,…,n, required by Definition 2, still maintains the liveness of each process subnet N_i with $M(p_{0i}) \neq M_0(p_{0i})$. Hence, the occurrence of the system deadlock at $\underline{M}$ must be caused by insufficiently marked resource places.

Finally, part (iii) of Lemma 1 is an immediate consequence of the above definition of set S. ♦

In the following, a deadly marked siphon S satisfying also the conditions (ii) and (iii) in Lemma 1, will be called a *resource-induced* deadly marked siphon. Lemma 1 essentially specializes the well-established connection in general PN theory between total deadlocks and badly marked siphons to the considered class of resource-process nets. As it is shown in the sequel, it provides a vehicle for connecting the non-liveness and/or non-quasi-liveness arising in this RAS class to resource-induced deadly marked siphons, by establishing that the lack of (any of)

these properties implies the existence of a reachable marking $M \neq M_0$ with a total deadlock in the modified marking $\underline{M}$. In this way, it reveals and establishes the natural connection between the concept of deadlock in the classical resource allocation system theory and its counterpart in the PN theory. The investigation of the implications of this connection for even broader classes of resource allocation systems modeling modern workflows, is undertaken in [6].

Lemma 2: Consider a G-AMG net $N=(P_S \cup P_I \cup P_F \cup P_0 \cup P_{R,}, T_S \cup T_I \cup T_F, W, M_0)$. If N is not quasi-live, then, $\exists M \in R(N,M_0)$ s.t. $M \neq M_0$ and the corresponding modified marking $\underline{M}$ is a total deadlock.

Proof: Since N is not quasi-live, there exists a transition t* that is dead at M_0. Let t* belong to the process subnet N_{i*}. From Definition 1, it is clear that $t^* \neq t_{Ii*}$, since t_{Ii*} is enabled at M_0. Let M' denote the marking of N reached from M_0 by firing t_{Ii*} once. Then, the fact that the subnet obtained from N_{i*} with the removal of its idle place p_{0i*} is a connected acyclic marked graph with t_{Ii*} as its unique source node, implies that every transition sequence σ s.t. $M'[\sigma>$ and $\forall t \in \sigma$, $t \in T_{Si*} \cup \{t_{Fi*}\}$, has finite length. Consider such a maximal transition sequence σ^* and let $M'[\sigma^*>M$. The deadness of t* implies that t_{Fi*} is not included in σ^* (c.f. [8], Prop. 3.16). Therefore, $M(p_{0i}) \neq M_0(p_{0i})$. Furthermore, $\underline{M}$ is a total deadlock for N, since $\forall i \neq i^*$, N_i is empty of tokens, and therefore, all its transitions are dead, while the deadness of the transitions in N_{i*} is established by the maximality of σ^*. ♦

Lemma 3: Consider a quasi-live G-AMG net $N=(P_S \cup P_I \cup P_F \cup P_0 \cup P_{R,}, T_S \cup T_I \cup T_F, W, M_0)$. If N is not live, then, $\exists M \in R(N,M_0)$ s.t. $M \neq M_0$ and the corresponding modified marking $\underline{M}$ is a total deadlock.

Proof: Since N is not live, $\exists M' \in R(N,M_0)$ and $t^* \in T$ s.t. t* is dead at M'. We claim that $\exists M \in R(N,M')$ s.t. $M \neq M_0$ and every transition $t \notin T_I$ is disabled. Indeed, the structure of the process subnets N_i, i=1,…,n, specified by Definition 1, implies that every transition sequence σ s.t.(i) $M'[\sigma>$ and (ii) for all $t \in \sigma$, t is not in T_I, is of finite length. Consider such a maximal transition sequence σ^* and let $M'[\sigma^*>M$. Then, $M \neq M_0$, since otherwise the quasi-liveness of N implies that t* is not dead at M'. To see that $\underline{M}$ is a total deadlock for N, simply notice that the specification of $\underline{M}$, by setting $M(p_{0i}) = 0$, $\forall i$, essentially disables all transitions $t \in T_I$, that are the only transitions potentially enabled in M. ♦

Theorem 1: Let $N=(P_S \cup P_I \cup P_F \cup P_0 \cup P_{R,}, T_S \cup T_I \cup T_F, W, M_0)$ be a marked G-AMG net. N is live if and only if the space of modified reachable markings, $\underline{R(N,M_0)}$ contains no resource-induced deadly marked siphons.

Proof: To show the necessity part, suppose that there exists $M \in R(N,M_0)$ s.t. $\underline{M}$ contains a resource-induced deadly marked siphon S. Let $r \in S \cap P_R$ be one of the disabling resource places, and consider $t \in r^\bullet$ s.t. $\underline{M}(r) < W(r,t)$. The definition of deadly marked siphon implies that $\forall t' \in {}^\bullet r$, t' is dead in $R(N,\underline{M})$. Furthermore, Definition 2 also implies that $\forall M' \in R(N,M)$, $M'(r) \leq M(r) = \underline{M}(r)$, since the re-introduction of the removed tokens in places $p \in P_0$, and their potential loading into the system, can only decrease the resource availabilities. Therefore, t is a dead transition at M, which contradicts the assumption of the net liveness.

To show the sufficiency part, suppose that N is not live. Then, the combination of Lemmas 2 and 3 imply that there exists $M \in R(N,M_0)$ s.t. $M \neq M_0$ and the corresponding modified marking $\underline{M}$ is a total deadlock. But then, Lemma 1 implies that $\underline{R(N,M_0)}$ contains a resource-induced deadly marked siphon, which contradicts the working hypothesis. ♦

B. Practical Implications of Theorem 1

From a more practical standpoint, the main concern in the control of contemporary RAS is that all activated processes can proceed to completion and get unloaded from the system without external intervention, i.e., that the running processes do not get entangled in *deadlock* or *livelock*. Given the characterization of the G-AMG initial state M_0, in item 7 of Definition 1, the above requirement for deadlock and livelock-free operation essentially translates into a requirement for *reversibility*, in the PN modeling framework. It should be obvious, however, that in the context of G-AMG nets, liveness is a necessary condition for reversibility. The next theorem establishes that liveness is also a sufficient condition for reversibility, and therefore, in the considered context, these two notions are equivalent.

Theorem 2: In the class of G-AMG nets, liveness implies reversibility.

Proof: Consider a live G-AMG net N, and a marking $M \in R(N,M_0)$ s.t. $M \neq M_0$. Then, using an argument similar to that used in the proof of Lemma 3, one can construct a finite-length firing sequence σ leading to a marking M' s.t. every transition $t \notin T_I$ is disabled in M'. We claim that $M'=M_0$. Indeed, by construction, $\underline{M}$' is a total deadlock of N, and if $M' \neq M_0$, Lemma 1 implies that M' contains a resource-induced deadly marked siphon. But then, Theorem 1 implies that N is not live, which contradicts the working hypothesis. ♦

As it was pointed out in the Introduction, it is expected that by connecting the net liveness and reversibility to the absence of resource-induced deadly marked siphons, the results of Theorems 1 and 2 will facilitate the synthesis of *liveness-enforcing supervisors (LES)* for quasi-live G-AMG's through the analytical methodology originally developed in [5] for the liveness-enforcing supervision of C/D-RAS. The key requirement for the application of this methodology in the A/D-RAS context, is the identification of a LES class that (i) will admit a PN-based representation;

(ii) when superimposed on the original uncontrolled RAS-modelling PN, it will preserve the structure of the G-AMG net;[3] (iii) for the case of G-AMG's with quasi-live process sub-nets, it will guarantee the structural liveness of the controlled system with respect to marking of the superimposed control subnet, i.e., the existence of a marking for the superimposed control places such that the resulting controlled net is live. Assuming the existence of such a LES class, a correct supervisor can be obtained by searching the marking space of the control subnet for a marking that does not give rise to resource-induced deadly marked siphons in the modified reachability space of the controlled system. This search can be based on a mathematical (integer) programming formulation that was developed in [5]; we refer the reader to that work for the further details.

Finally, the combination of Lemmas 1 and 2 implies that, in principle, the aforementioned computational technique of [5] can provide also an analytical sufficiency test for the quasi-liveness of any given G-AMG process net. The resolving power of this test will be rather limited, but still its availability is quite important, since it has been already established in [9] that evaluating the quasi-liveness of a G-AMG process net is an NP-complete problem.

IV. CONCLUSION

The work presented in this paper first introduced the PN class of Generalized Augmented Marked Graphs (G-AMG's) for modeling the behavior of RAS with synchronization constraints, and subsequently it established that the lack of liveness and/or quasi-liveness in this new PN class can still be attributed to the development of resource-induced deadly marked siphons in the modified reachability space. In this way, it extended past similar results concerning the structural characterization of liveness in PN's modeling the behavior of C/D-RAS. From a more practical standpoint, it was highlighted how the derived results can provide the basis for (i) the development of a methodology for verifying the liveness and quasi-liveness of G-AMG nets, and (ii) the synthesis of LES that are appropriate for the A/D-RAS operational context, by building upon some computational tools developed in [5].

Future work will seek (i) to fully develop the aforementioned methodology regarding the evaluation of G-AMG liveness and quasi-liveness, and the synthesis of the necessary LES, and (ii) to extend the siphon-based characterization of non-liveness and deadlock derived in this work, to RAS classes with more complex behavioral patterns. Some initial results towards the latter direction are reported in [6].

V. ACKNOWLEDGEMENTS

This work was partially supported by the Keck Foundation.

VI. REFERENCES

[1] S. Reveliotis, Liveness Enforcing Supervision for Sequential Resource Allocation Systems: State of the Art and Open Issues, *in Synthesis and Control of Discrete Event Systems,* by B. Caillaud, X. Xie, P. Darondeau and L. Lavagno (eds.), Kluwer Academic Pub., 2002, pp 203-212.

[2] J. Ezpeleta, J. M. Colom and J. Martinez, A Petri net Deadlock Prevention Policy for Flexible Manufacturing Systems, *IEEE Trans. on R&A*, Vol. 11, 1995, pp 173-184.

[3] F. Chu and X. Xie, Deadlock Analysis of Petri nets using Siphons and Mathematical Programming, *IEEE Trans. on R&A,* Vol. 13, 1997, pp 793-804.

[4] X. Xie and M. Jeng, ERCN-merged Nets and their Analysis using Siphons, *IEEE Trans. on R&A,* Vol. 15, 1999, pp 692-703.

[5] J. Park and S. Reveliotis, Deadlock Avoidance in Sequential Resource Allocation Systems with Multiple Resource Acquisitions and Flexible Routings, *IEEE Trans. on Automatic Control*, Vol. 46, 2001, pp 1572-1583.

[6] S. Reveliotis, On the Siphon-based Characterization of Liveness in Sequential Resource Allocation Systems, *Tech. Report*, School of Industrial & Systems, Eng. Georgia Tech, 2001.

[7] M. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach,* World Scientific, Singapore, 1998.

[8] J. Desel and J. Esparza, *Free Choice Petri nets,* Cambridge University Press, 1995.

[9] E. Roszkowska and R. Wojcik, Problems of Process Flow Feasibility in FAS, in *CIM in Process and Manufacturing Industries,* Pergamon Press, 1993, pp 115-120.

[3] or, any other PN class in which non-liveness can be attributed to the presence of (resource-induced) deadly marked siphons…