Siphon-based characterization of liveness and liveness-enforcing supervision for sequential resource allocation systems

Spyros A. Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology
765 Ferst Drive
Atlanta, GA 30332
phone: (404) 894-6608, fax: (404) 894-2301, e-mail: *spyros@isye.gatech.edu*

Abstract— **One of the most interesting developments from, both, a theoretical and a practical perspective, in the emerging theory of resource allocation systems (RAS), is the characterization of the non-liveness of many RAS classes through the Petri net (PN)-based structural object of empty, or more generally, deadly marked siphon. The work presented in this paper seeks to develop a general theory that provides a unifying framework for all the relevant existing results, and reveals the key structures and mechanisms that connect the RAS non-liveness to the concept of deadly marked – and in certain cases, empty – siphon. In this capacity, the presented results allow also the extension of the siphon-based characterization of non-liveness to broader RAS classes, and provide a clear and intuitive explanation for the limitations of the approach. The last part of the work discusses how the derived structural characterization of RAS non-liveness can be combined with some algorithms for detecting empty or deadly marked siphons in a given PN marking, in order to develop analytical liveness sufficiency tests and systematic procedures for the design of liveness-enforcing supervisors (LES).**

Keywords: sequential resource allocation systems, deadlock resolution, Petri net structural analysis, siphons

I. Introduction

One of the major breakthroughs underlying our capability to systematically evaluate the liveness of various resource allocation system (RAS) configurations, and to synthesize effective and computationally efficient liveness-enforcing supervisors for non-live RAS, is the formal characterization of the non-liveness of the Petri net (PN) sub-classes modelling the behavior of these environments, through the formation of a particular PN structural object, known as *empty* or, more generally, *deadly marked siphon*[1] [1]. This type of results can be originally traced in the seminal work of Ezpeleta and his colleagues [2], that provided a siphon-based characterization for (non-)liveness in *Single Unit (SU)* RAS [3], i.e., a sequential RAS subclass in which processes execute in (partially) ordered sequences of stages, with each stage requiring for its support the exclusive allocation of one unit from a single resource type. More specifically, [2] established that in the class of SU-RAS, non-liveness can be interpreted through the development of *empty* siphons in the system reachability space. Similarly, the work of [4] established that the development of reachable empty siphons is also the cause for non-liveness in *Augmented Marked Graphs (AMG)*, a RAS model that generalizes the class of SU-RAS by introducing task parallelism, through the presence of merging and splitting transitions. The role of empty siphons for the non-liveness of single-unit RAS with merging and splitting transitions was subsequently investigated more extensively in [5]. At the same time, the work of [6] established that the behavior of an SU-RAS under the control of a class of liveness-enforcing supervisors (LES) expressed by a set of linear inequality constraints on the RAS state – known as *algebraic* LES – can be modelled as an AMG, and therefore, the aforementioned results of [4] provide a structural test for assessing the LES correctness. Beyond its theoretical value, this finding is of considerable practical significance since the work of [4] provides also computational sufficiency tests for the non-existence of reachable empty siphons, that take the convenient form of mathematical programming (MP) formulations polynomially sized, in terms of variables and constraints, with respect to the underlying PN model. More recently, this entire set of results, originally developed in the context of SU-RAS, has been generalized to the broader RAS class involving *Conjunctive (C)* [3] resource allocation, i.e., to RAS environments that engage an arbitrary set of resources, each of them at an arbitrary level of units, for the support of a single process stage [7], [8]. In the C-RAS operational context, the new PN structural object characterizing the non-liveness of the RAS behavior is that of *deadly marked* siphon, while its presence is detected on a *modified* reachability space, that constitutes a projection of the original PN reachability space to a subspace defined by a subset of the components of the net marking vector [7]. The work in [7] developed also MP-based sufficiency tests for the non-existence of deadly marked siphons in the aforementioned modified reachability space, and therefore, it offers computational tools for developing algebraic LES for this broader class of RAS behavior. Finally, some additional works that have investigated the role of the PN siphon structure in the liveness of sequential resource allocation, are those presented in [9], [10], [11], [12], [13].

As it is evident from the above discussion, currently,

[1] All technical concepts are systematically introduced in the later parts of this paper.

all the existing results on siphon-based characterization of liveness in sequential RAS have been developed in a rather fragmented fashion, each of them pertaining to a RAS sub-class characterizing a particular type of RAS behavior. Furthermore, the detailed study of their development would reveal that, while they are based on formal and rigorous technical arguments, they fail to provide an explicit and intuitive characterization of the underlying key mechanism that links the non-liveness of the considered RAS classes to the presence of some empty or deadly marked siphons. Hence, the work presented in this paper seeks to develop a general theory that

- will provide a systematic explanation of the relationship between the RAS non-liveness and the presence of deadly marked – and in the case of SU-RAS, empty – siphons;
- will offer, thus, a unifying framework for interpreting all the relevant results existing in the literature;
- will allow the extension of the existing results to broader and/or other RAS behaviors;
- will eventually enable a systematic methodology for establishing correct and live behavior in the considered RAS classes, through a *decomposition* that differentiates between (a) the design of the involved process flows, and (b) the synthesis of a supervisor that manages the allocation of the system resources to the concurrently executing process instances in a way that preserves the system liveness.[2]

Our approach is based on the identification of a minimal set of requirements for the structure of the RAS processes and their behavior, which when met, will allow the attribution of any experienced RAS non-liveness to the development of deadly marked siphons in the modified reachability space of the RAS-modelling PN. Two concepts that are shown to play a central role in this minimal set of requirements are those of the process *quasi-liveness* and *reversibility*. These properties essentially imply that the execution logic underlying the various process flows is inherently consistent, and therefore, any non-liveness of the PN modelling the overall RAS behavior can be attributed to the competition of the concurrently executing processes for the finite system resources. A third requirement that appears in the subsequent results, and it is necessary in order to connect the non-liveness of the process-resource net to the presence of deadly marked siphons, is that of *acyclic process flows*, i.e., the developed results pertain to RAS in which the various processes do not present re-circulating loops among their different stages.[3]

The rest of the paper is organized as follows: Section 2 first presents the PN fundamentals that are necessary for the modelling and analysis of the considered RAS structure and behavior, and subsequently it proceeds to the systematic characterization of this RAS class through a series of definitions and assumptions. Section 3 develops the main structural results of this work, by establishing that for RAS with quasi-live, reversible and acyclic processes, non-liveness can be attributed to the development of deadly marked siphons in the modified reachability space of the RAS-modelling PN. It also indicates how all the currently existing results connecting liveness to empty or deadly marked siphons, can be obtained as special cases of this more general development. Moreover, the presented formal argument establishing the connection between non-liveness and deadly marked siphons provides also an intuitive explanation for it, since it reveals that, in the considered RAS class, non-liveness can be attributed to the formation of *total* deadlocks in the modified PN reachability space. Finally, Section 3 establishes that in the considered class of systems, liveness and reversibility are equivalent concepts, providing, thus, the formal link between the concept of PN liveness and the typical concerns of RAS deadlock avoidance theory. Section 4 overviews the MP-based sufficiency test for the non-existence of deadly marked siphons in the underlying modified reachability space, originally developed in [7], and it discusses how this test can support the synthesis of correct algebraic LES for any instance of the considered RAS class. Finally, Section 5 concludes the paper, and identifies some additional research issues originating from the presented work.

II. The considered RAS class and its Petri net model

This section first overviews the Petri net (PN) related concepts that are necessary for the formal modelling of the considered RAS class and the analysis of its properties, and subsequently, it provides a detailed characterization of the PN structure modelling the considered resource allocation environments. Some excellent more extensive treatments of the PN modelling framework and the structural and behavioral analysis of PN models can be found in [17], [18].

A. Petri net preliminaries

A *marked Petri Net (PN)* is defined by a quadruple $\mathcal{N} = (P, T, W, M_0)$, where P is the set of *places*, T is the set of *transitions*, $W : (P \times T) \cup (T \times P) \to Z^+$ is the *flow relation*, and $M_0 : P \to Z^+$ 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*. If only the restriction of W to $(P \times T)$ maps on $\{0, 1\}$, the PN is said to be *PT-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. $^\bullet X = \bigcup_{x \in X} {}^\bullet x$. The ordered set $X =< x_1 \ldots x_n > \in \ (P \cup T)^*$ is a *path*, if and only if (iff) $x_{i+1} \in x_i^\bullet, i = 1, \ldots, n-1$. Furthermore, a path X is characterized as a *circuit* 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*).

[2] It must be mentioned at this point, that while the present paper was in the review process, reference [14] was published, with a set of results quite similar to the research program outlined above. It is emphasized that these two works were developed simultaneously and independently, and throughout the subsequent development, we point out the similarities and differences among them.

[3] This requirement can be relaxed under certain conditions; c.f. [15], [16] for details.

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 $M[t\rangle$. $t \in T$ is said to be *disabled* by $p \in {}^\bullet t$ 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 $R(\mathcal{N}, M_0)$. A marked PN $\mathcal{N}$ with initial marking M_0 is said to be *bounded* iff all markings $M \in R(\mathcal{N}, M_0)$ are bounded, while $\mathcal{N}$ is said to be *structurally bounded* iff it is bounded for any initial marking M_0. $\mathcal{N}$ is said to be *reversible* iff $\forall M \in R(\mathcal{N}, M_0)$, $M_0 \in R(\mathcal{N}, M)$.

In case that a marked PN is *pure* (i.e., $\forall (x,y) \in (P \times T) \cup (T \times P)$, $W(x,y) > 0 \Rightarrow W(y,x) = 0$), 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* y is a $|P|$-dimensional vector satisfying $y^T\Theta = 0$ and $y \geq 0$, and a *t-semiflow* x is a $|T|$-dimensional vector satisfying $\Theta x = 0$ and $x \geq 0$. A p-semiflow y (t-semiflow x, resp.) is said to be *minimal* iff $\not\exists$ a p-semiflow y' (t-semiflow x', resp.) such that $\|y'\| \subset \|y\|$ ($\|x'\| \subset \|x\|$, resp.), where $\|y\| = \{p \in P \mid y(p) > 0\}$ ($\|x\| = \{t \in T \mid x(t) > 0\}$, resp.).

Given a marked PN $\mathcal{N} = (P, T, W, M_0)$, a transition $t \in T$ is *live* iff $\forall M \in R(\mathcal{N}, M_0), \exists M' \in R(\mathcal{N}, M)$ s.t. $M'[t\rangle$, and $t \in T$ is *dead* at $M \in R(\mathcal{N}, M_0)$ iff $\not\exists$ marking $M' \in R(\mathcal{N}, M)$ s.t. $M'[t\rangle$. A marking $M \in R(\mathcal{N}, M_0)$ is a (total) *deadlock* iff $\forall t \in T$, t is dead. A marked PN $\mathcal{N}$ is *quasi-live* iff $\forall t \in T, \exists M \in R(\mathcal{N}, M_0)$ s.t. $M[t\rangle$, it is *weakly live* iff $\forall M \in R(\mathcal{N}, M_0), \exists t \in T$ s.t. $M[t\rangle$, 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 $S \subseteq P$ such that ${}^\bullet S \subseteq S^\bullet$. A siphon S is *minimal* iff $\not\exists$ a siphon S' s.t. $S' \subset S$. A siphon S is said to be *empty* at marking M iff $M(S) \equiv \sum_{p \in S} M(p) = 0$, and it is said to be *deadly marked* at marking M, iff $\forall t \in {}^\bullet S$, t is disabled by some $p \in S$ [7]. Obviously, empty siphons are deadly marked siphons. It is easy to see that, 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) $\forall M' \in R(\mathcal{N}, M)$, S is deadly marked. Furthermore, it can be shown that if marking $M \in R(\mathcal{N}, M_0)$ is a total deadlock, then the set S of disabling places in M constitutes a deadly marked siphon [7]. This last result constitutes the generalization of a well-established relationship between total deadlocks and empty siphons in ordinary PN's [18].

Finally, given two PN's $\mathcal{N}_1 = (P_1, T_1, W_1, M_{01})$ and $\mathcal{N}_2 = (P_2, T_2, W_2, M_{02})$ with $T_1 \cap T_2 = \emptyset$ and $P_1 \cap P_2 = Q \neq \emptyset$ s.t. $\forall p \in Q$, $M_{01}(p) = M_{02}(p)$, the PN $\mathcal{N}$ resulting from the *merging* of the nets $\mathcal{N}_1$ and $\mathcal{N}_2$ *through the place set* Q, is defined by $\mathcal{N} = (P_1 \cup P_2, T_1 \cup T_2, W_1 \cup W_2, M_0)$ with $M_0(p) = M_{01}(p)$, $\forall p \in P_1 \backslash P_2$; $M_0(p) = M_{02}(p)$, $\forall p \in P_2 \backslash P_1$; $M_0(p) = M_{01}(p) = M_{02}(p)$, $\forall p \in P_1 \cap P_2$.

B. The considered RAS class and the associated PN model

For the purposes of the liveness analysis considered in this work, a *(sequential) resource allocation system (RAS)* is formally defined by a set of *resource types* $\mathcal{R} = \{R_l,\ l = 1, \ldots, m\}$, each of them available at some finite *capacity* $C_l \in Z^+$, and a set of *process types* $\mathcal{J} = \{J_j,\ j = 1, \ldots, n\}$, that execute sequentially, through a number of *tasks* or *stages*, $J_{jk}, k = 1, \ldots, \lambda_j$, and with each stage J_{jk} engaging a specific subset of the system resources for its execution. More specifically, it is assumed that a process instance advances to the execution of a certain stage, J_{jk}, only after it has secured the required resources, and upon its advancement, it releases the resources held for the execution of the previous stage $J_{j,k-1}$. Furthermore, the set of *tasks* or *stages*, $\{J_{jk},\ k = 1, \ldots, \lambda_j\}$, corresponding to process type J_j, presents some additional structure that expresses the associated *process-defining logic* and characterizes the potential process *routings*. Typical structures involved in the definition of the process logic include linear, parallel, conditional and iterative structures, as well as more complex structures resulting from the nested combination of the basic ones. Most of the past research on RAS liveness and liveness-enforcing supervision has focused on simpler process structures that allow the modelling of simple linear process flows, potentially enhanced with some routing flexibility (e.g., [2], [19], [20], [21], [22], [7]).

This work does not make any explicit assumptions about the specific structure of the considered RAS processes, but it only requires that the involved process logic is "inherently consistent", and therefore, any non-liveness arising in the behavior of the resulting RAS and its associated PN model can be attributed to the (mis-)management of the allocation of the finite set of the system resources to the concurrently executing processes. A formal characterization of this notion of *"inherent process consistency"* is provided by the following definition of the considered *process subnet* and its assumed properties.

Definition 1: For the purposes of this work, a *process (sub-)net* is a Petri net $\mathcal{N}_P = (P, T, W, M_0)$ such that:

i. $P = P_S \cup \{i,\ o\}$ with $P_S \neq \emptyset$;
ii. $T = T_S \cup \{t_I,\ t_F,\ t^*\}$;
iii. $i^\bullet = \{t_I\}$; ${}^\bullet i = \{t^*\}$;
iv. $o^\bullet = \{t^*\}$; ${}^\bullet o = \{t_F\}$;
v. $t_I^\bullet \subseteq P_S$; ${}^\bullet t_I = \{i\}$;
vi. $t_F^\bullet = \{o\}$; ${}^\bullet t_F \subseteq P_S$;
vii. $(t^*)^\bullet = \{i\}$; ${}^\bullet(t^*) = \{o\}$;
viii. the underlying digraph is *strongly connected*;
ix. $M_0(i) > 0 \ \wedge \ M_0(p) = 0,\ \forall p \in P \backslash \{i\}$;
x. $\forall M \in R(\mathcal{N}_P, M_0)$, $M(i) + M(o) = M_0(i) \Longrightarrow M(p) = 0,\ \forall p \in P_S$.

□

In the PN-based process representation introduced by Definition 1, process instances waiting to initiate processing are represented by tokens in place i, while the initiation of a process instance is modelled by the firing of transition t_I. Similarly, tokens in place o represent completed process instances, while the event of a process completion is modelled by the firing of transition t_F. Transition t^* allows the token re-circulation – i.e., the token transfer from place o

to place i – in order to model *repetitive* process execution. Finally, the part of the net between transitions t_I and t_F that involves the process places P_S, models the sequential logic defining the considered process type, and, as it can be seen in Definition 1, it can be quite arbitrary. However, in order to capture the notion of the "inherent process consistency" introduced at the beginning of this sub-section, we further qualify the considered process sub-nets through the following two assumptions:

Assumption 1: The process (sub-)nets considered in this work are assumed to be *quasi-live* for $M_0(i) = 1$.
□

Assumption 2: The process (sub-)nets considered in this work are assumed to be *reversible* for *every* initial marking M_0 that satisfies Condition (ix) of Definition 1.
□

Assumption 1 stipulates that the every transition in the considered process sub-net models a meaningful event that can actually occur during the execution of some process instance, and therefore, it is not redundant. On the other hand, Assumption 2 essentially stipulates that, at any point in time, all *active* process instances can proceed to completion, and this completion can occur without the initiation of any additional process instances.[4] When taken together, Assumptions 1 and 2 imply also the *liveness* of the considered process nets; we state this result as a lemma, but we skip its proof, since it is a rather well-known result in the PN-research community.

Lemma 1: Under Assumptions 1 and 2, the considered process nets are also *live*.
□

Since the emphasis of this work is on the characterization and establishment of live resource allocation, the complete characterization of the class of process nets satisfying Assumptions 1 and 2 lies beyond its scope. We notice, however, that all the RAS classes for which there exist results connecting their non-liveness to the development of deadly marked / empty siphons, involve PN-based process models that satisfy the aforementioned assumptions.

Another assumption that is necessary for the development of the analytical results of the next section, is that the various process (sub-)nets are *acyclic*. This concept is defined as follows:

Assumption 3: The process sub-nets considered in this work are assumed to be *acyclic*, i.e., the removal of transition t^* from them renders them acyclic digraphs.
□

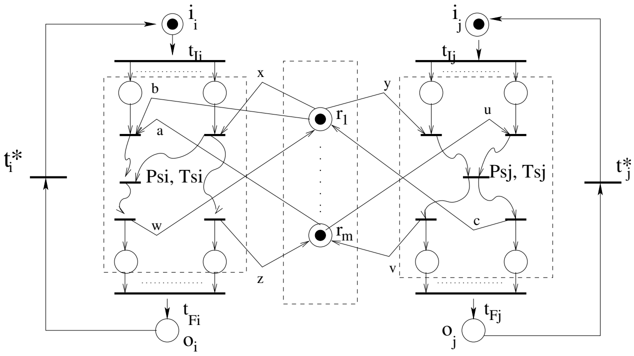

Fig. 1. The process-resource net structure considered in this work

The modelling of the resource allocation associated with each process stage, $p \in P_S$, necessitates the augmentation of the process sub-net $\mathcal{N}_P$, defined above, with a set of *resource* places $P_R = \{r_l,\ l = 1, \ldots, m\}$, of initial marking $M_0(r_l) = C_l,\ i = 1, \ldots, m$, and with the corresponding flow sub-matrix, Θ_{P_R}, expressing the allocation and de-allocation of the various resources to the process instances as they advance through their processing stages. Notice that the interpretation of the role of transitions t^*, t_I and t_F implies that $(t^*)^\bullet \cap P_R = {}^\bullet(t^*) \cap P_R = (t_I)^\bullet \cap P_R = {}^\bullet(t_F) \cap P_R = \emptyset$. The resulting net will be called the *resource-augmented process (sub-)net* and it will be denoted by $\overline{\mathcal{N}_P}$. The *reusable* nature of the system resources is captured by the following assumption regarding the resource-augmented process net $\overline{\mathcal{N}_P}$:

Assumption 4: Let $\overline{\mathcal{N}_P} = (P_S \cup \{i,\ o\} \cup P_R, T, W, M_0)$ denote a resource-augmented process (sub-)net. Then, $\forall l \in \{1, \ldots, |P_R|\}$, there exists a p-semiflow y_{r_l}, s.t.: (i) $y_{r_l}(r_l) = 1$; (ii) $y_{r_l}(r_j) = 0,\ \forall j \neq l$; (iii) $y_{r_l}(i) = y_{r_l}(o) = 0$; (iv) $\forall p \in P_S,\ y_{r_l}(p) =$ number of units from resource R_i required for the execution of stage p.
□

While the p-semiflows introduced by Assumption 4 characterize the resource allocation taking place at each process stage and the conservative nature of the system resources, they do not reveal anything regarding the adequacy of the available resource set for supporting the execution of the various processing stages, under the sequencing constraints implied by the process-defining logic. This additional concern underlying the correct definition of the various RAS process-types is captured by extending the requirement for *quasi-liveness* of the process net $\mathcal{N}_P$, introduced by Assumption 1, to the resource-augmented process net $\overline{\mathcal{N}_P}$:

Assumption 5: The resource-augmented process (sub-)nets considered in this work are assumed to be *quasi-live* for $M_0(i) = 1$ and $M_0(r_l) = C_l,\ \forall l \in \{1, \ldots, |P_R|\}$.
□

The complete PN-based model, $\mathcal{N} = (P, T, W, M_0)$, of any given instance from the considered RAS class is obtained by *merging* the resource-augmented process nets $\overline{\mathcal{N}_{P_j}} = (P_j, T_j, W_j, M_{0_j}),\ j = 1, \ldots, n$, modelling its constituent process types, through their common resource places. The resulting PN class is characterized as the class of *process-resource nets with quasi-live, reversible and*

[4] It is noticed, for completeness, that the requirement for process reversibility introduced by Assumption 2 when combined with Definition 1 and Assumption 1, subsumes the notion of process *soundness*, introduced in Workflow theory (c.f., [23], [24]) in order to characterize well-defined process (sub-)nets, for the case where only a single process instance re-circulates in the considered process net. However, Assumption 2 further stipulates that when more than one process instances have been activated, still they will always be able to complete, in spite of any additional effects arising from their interaction through the defining process logic. This requirement plays also an important role in the developments presented in [14]; specifically, in [14], the authors introduce the term *"strong reversibility"* in order to characterize the requirement expressed by Assumption 2 as an additional net property.

acyclic process sub-nets, and its basic structure is depicted in Figure 1. Let $P = \bigcup_j P_j$, $P_S = \bigcup_j P_{S_j}$; $I = \bigcup_j \{i_j\}$; $O = \bigcup_j \{o_j\}$. Then, $P = P_S \cup I \cup O \cup P_R$. Furthermore, the re-usable nature of the resource allocation taking place in the entire process-resource net is characterized by a p-semiflow y_{r_l} for each resource type R_l, $l = 1, \ldots, m$, defined by: (i) $y_{r_l}(r_l) = 1$; (ii) $y_{r_l}(r_j) = 0$, $\forall j \neq l$; (iii) $y_{r_l}(i_j) = y_{r_l}(o_j) = 0$, $\forall j$; (iv) $\forall p \in P_S$, $y_{r_l}(p) = y_{r_l}^{(j*)}(p)$, where $\overline{\mathcal{N}_{P_{j*}}}$ denotes the resource-augmented process sub-net containing place p, and $y_{r_l}^{(j*)}()$ denotes the corresponding p-semiflow for resource R_l. Finally, it is easy to see that Assumption 5 regarding the quasi-liveness of the constituent resource-augmented process sub-nets $\overline{\mathcal{N}_{P_j}}$ implies also the quasi-liveness of the entire process-resource net $\mathcal{N}$.

The next definition extends to the class of process-resource nets, considered in this work, the notion of the *modified* marking, originally introduced in [25], [7] for analyzing the liveness of a more restricted PN subclass, modelling the behavior of sequential RAS with multi-unit resource allocation per stage and routing flexibility.

Definition 2: Given a process-resource net $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ and $M \in R(\mathcal{N}, M_0)$, the *modified marking* $\overline{M}$ is defined by

$$\overline{M}(p) = \begin{cases} M(p) & \text{if } p \notin I \cup O \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Furthermore, the set of all modified markings induced by the net reachable markings is defined by $\overline{R(\mathcal{N}, M_0)} = \{\overline{M} \mid M \in R(\mathcal{N}, M_0)\}$
□

We conclude this section by noticing that, from a completely practical standpoint, the main requirement underlying the classical RAS deadlock avoidance theory, is that every process activated in the system will be able to run to completion, without getting entangled in a deadlock situation [26]. In the PN-modelling framework, this requirement is explicitly modelled by stipulating that the corresponding process-resource net is *reversible*. However, in the considered class of process-resource nets, reversibility is equivalent to liveness. This result is established in Section 3, and justifies, in terms of the more typical requirements of the RAS deadlock avoidance theory, the overall approach taken in this work.

III. Liveness analysis of process-resource nets with quasi-live, reversible and acyclic process sub-nets

The main result of this section links the non-liveness arising in the class of process-resource nets with quasi-live, reversible and acyclic process sub-nets,[5] to the development of a special type of deadly marked siphon in the net modified reachability space. It is also discussed how this new result encompasses and explains all existing similar results, pertaining to more restricted RAS classes. The last part of the section establishes that, for the considered class of process-resource nets, liveness and reversibility are equivalent concepts.

The result connecting the non-liveness arising in the class of process-resource nets with quasi-live, reversible and acyclic process sub-nets to the presence of deadly marked siphons is developed in a three-step argument, that further reveals the fundamental structures and mechanisms behind it. Hence, its derivation provides also the intuitive explanation requested in the opening discussion of this paper. The first step in this development is established by the following lemma:

Lemma 2: Consider a process-resource net $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ with quasi-live and reversible process sub-nets. If $\exists M \in R(\mathcal{N}, M_0)$ s.t. $\exists$ a process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$ and $\overline{M}$ is a total deadlock, then $\exists$ siphon S s.t.

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

Proof: Let S denote the set of disabling places in modified marking $\overline{M}$. Since $\overline{M}$ is a total deadlock, $S^\bullet = T \supseteq {}^\bullet S$. Therefore, S is a siphon, while the definition of S implies also that it is deadly marked. This establishes part (i) in the above lemma.

To establish that $S \cap P_R \neq \emptyset$, consider the process sub-net $\mathcal{N}_{P_{j^*}}$. The fact that $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$ implies that there are active process instances in the sub-net $\mathcal{N}_{P_{j^*}}$. But then, Assumptions 2 and 1 imply that sub-net $\mathcal{N}_{P_{j^*}}$ remains live in spite of any token removal from places i_{j^*} and o_{j^*} requested by Definition 2. Hence, the occurrence of the system deadlock at $\overline{M}$ must involve insufficiently marked resource places.

Finally, part (iii) of Lemma 2 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 2, will be called a *resource-induced* deadly marked siphon. Lemma 2 essentially specializes the more general connection between total deadlocks and deadly marked siphons (c.f., Section 2.1), to the subclass of process-resource nets with quasi-live and reversible active processes. From a methodological standpoint, it provides a vehicle for connecting the liveness of resource allocation – and, in certain cases, even the quasi-liveness – taking place in process-resource nets, to resource-induced deadly marked siphons, as long as it can be established that the lack of (any of) these properties implies the existence a reachable marking M s.t. (i) there exists a process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$ and (ii) the corresponding modified marking $\overline{M}$ is a total deadlock. The next lemma establishes that this is the case for the class of process-resource nets with quasi-live, reversible and acyclic process sub-nets.

Lemma 3: Consider a process-resource net $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ with quasi-live, reversible and acyclic process sub-nets. If $\mathcal{N}$ is not live, then, $\exists M \in R(\mathcal{N}, M_0)$ s.t. (i) $\exists$ process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$ and (ii) $\overline{M}$ is a total deadlock.

[5] We clarify that in the subsequent development of this paper, a process (sub-)net is characterized *quasi-live* if it satisfies Assumption 1 and the corresponding resource-augmented process sub-net satisfies Assumption 5, it is characterized *reversible* if it satisfies Assumption 2, and it is characterized *acyclic* if it satisfies Assumption 3.

Proof: Since $\mathcal{N}$ is not live, $\exists M' \in R(\mathcal{N}, M_0)$ and $t' \in T$ s.t. t' is dead in M'. We claim that $\exists M \in R(\mathcal{N}, M')$ s.t. (i) $\exists$ process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$ and (ii) every transition $t \notin (I \cup O)^\bullet$ is disabled in M. Indeed, the acyclic structure of the process sub-nets $\mathcal{N}_{P_j}$, $j = 1, \ldots, n$, implies that every transition sequence σ s.t. $M'[\sigma\rangle$ and $\forall t \in \sigma$, $t \notin (I \cup O)^\bullet$, will be of finite length. Consider such a maximal transition sequence $\hat{\sigma}$ and let $M'[\hat{\sigma}\rangle M$. Then, at marking M there must exist a process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$, since otherwise the initial marking M_0 is reachable from M, and then, the quasi-liveness of $\mathcal{N}$ implies that t' is not dead at M'. To see that $\overline{M}$ is a total deadlock for $\mathcal{N}$, simply notice that the specification of $\overline{M}$, by setting $\overline{M}(i_j) = \overline{M}(o_j) = 0$, $\forall j$, essentially disables all transitions $t \in (I \cup O)^\bullet$, that, by construction, are the only transitions potentially enabled in M. □

The next theorem completes the aforementioned three-step development of the key result of this section, by stating and proving, by means of Lemmas 2 and 3, that in the class of process-resource nets with quasi-live, reversible and acyclic processes, there is a direct relationship between the RAS non-liveness and the presence of resource-induced deadly marked siphons in the modified reachability space of the RAS-modelling PN.

Theorem 1: Let $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ be a process-resource net with quasi-live, reversible and acyclic processes. $\mathcal{N}$ is live if and only if the space of modified reachable markings, $\overline{R(\mathcal{N}, M_0)}$, contains no resource-induced deadly marked siphons.

Proof: To show the necessity part, suppose that $\exists M \in R(\mathcal{N}, M_0)$ s.t. $\overline{M}$ 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. $\overline{M}(r) < W(r,t)$. The definition of deadly marked siphon implies that $\forall t' \in {}^\bullet r$, t' is dead in $R(\mathcal{N}, \overline{M})$. This remark, when combined with Definition 2 and Assumption 4, further imply that $\forall M' \in R(\mathcal{N}, M)$, $M'(r) \leq M(r)$, since the re-introduction of the tokens removed from places $p \in I \cup 0$ and their potential loading in the system, can only decrease the resource availabilities. Therefore, t is a dead transition at M, which contradicts the assumption of net liveness.

To show the sufficiency part, suppose that $\mathcal{N}$ is not live. Then, Lemma 3 implies that $\exists M \in R(\mathcal{N}, M_0)$ s.t. (i) $\exists$ process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$, and (ii) $\overline{M}$ is a total deadlock. But then, Lemma 2 implies that $\overline{R(\mathcal{N}, M_0)}$ contains a resource-induced deadly marked siphon, which contradicts the working hypothesis. □

The following corollary results immediately from Theorem 1; its original statement (and a formal proof) can be found in [25], [7].

Corollary 1: Let $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ be a process-resource net where (i) the process sub-nets $\mathcal{N}_{P_j}$, $j = 1, \ldots, n$, are strongly connected state machines with each circuit containing the places i_j and o_j, and (ii) the resource-augmented process nets $\overline{\mathcal{N}_{p_j}}$ are quasi-live. Then, $\mathcal{N}$ is live if and only if the space of modified reachable markings, $\overline{R(\mathcal{N}, M_0)}$, contains no resource-induced deadly marked siphons.

□

The next corollary specializes Theorem 1 to the sub-class of process-resource nets where the process sub-nets $\mathcal{N}_{P_j}$ are acyclic marked graphs. A stronger version of this result, that connects also the lack of quasi-liveness to the presence of resource-induced deadly marked siphons, is presented in [8].

Corollary 2: Let $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ be a process-resource net where (i) the process sub-nets $\mathcal{N}_{P_j}$, $j = 1, \ldots, n$, are strongly connected marked graphs with each circuit containing the places i_j and o_j, and (ii) the resource-augmented process nets $\overline{\mathcal{N}_{p_j}}$ are quasi-live. Then, $\mathcal{N}$ is live if and only if the space of modified reachable markings, $\overline{R(\mathcal{N}, M_0)}$, contains no resource-induced deadly marked siphons.

□

The next result states that for the case of *PT-ordinary* PN's, the problematic siphons interpreting the RAS non-liveness are, in fact, *empty* siphons, and they can also be identified in the *original* net reachability space $R(\mathcal{N}, M_0)$ (besides the modified reachability space $\overline{R(\mathcal{N}, M_0)}$).

Corollary 3: Let $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ be a *PT-ordinary* process-resource net with quasi-live, reversible and acyclic process sub-nets. $\mathcal{N}$ is live if and only if the space of reachable markings, $R(\mathcal{N}, M_0)$, contains no empty siphons.

Proof: According to Theorem 1, under the assumptions of Corollary 3, net $\mathcal{N}$ is non-live, iff there exists a marking $M \in R(\mathcal{N}, M_0)$, s.t. $M \neq M_0$ and its modified marking $\overline{M}$ contains a resource-induced deadly marked siphon, S. Furthermore, the development of the result of Theorem 1 (c.f., Lemmas 2 and 3) indicates that S is defined by the set of disabling places of a total deadlock contained in $\overline{M}$. Since every place $p \in S$ is a disabling place in $\overline{M}$, and net $\mathcal{N}$ is PT-ordinary, $\overline{M}(p) = 0$, $\forall p \in S$. Hence, S is an empty siphon in $\overline{M}$. It remains to be shown that the presence of the resource-induced empty siphon S in the modified marking $\overline{M}$ implies the presence of an empty siphon S' in the original marking M. For that, let $S' = \{r_i : r_i \in S\} \cup \{p \in P_S : M(p) = \overline{M}(p) = 0 \ \wedge \ \exists r_i \text{ s.t. } (r_i \in S \wedge y_{r_i}(p) > 0)\}$. Notice that $S' \neq \emptyset$, since S is a resource-induced empty siphon. We show that S' is a siphon (which is empty, by construction), by considering the next two main cases:

Case I – $t \in {}^\bullet r_k$ **for some** $r_k \in S$**:** Then, $\exists q \in S$ s.t. $t \in q^\bullet$. If $q \in P_R$, then $q \in \{r_i : r_i \in S\} \subset S'$. On the other hand, if $q \notin P_R$, then $q \in P_S$, since $(q^\bullet)^\bullet \cap P_R \neq \emptyset$. Furthermore, $y_{r_k}(q) > 0$ and $M(q) = 0$ (since $q \in S$). Therefore, $q \in \{p \in P_S : M(p) = \overline{M}(p) = 0 \wedge \exists r_i \text{ s.t. } (r_i \in S \wedge y_{r_i}(p) > 0)\} \subset S'$. In both cases, $t \in (S')^\bullet$.

Case II – $t \in {}^\bullet q$ **for some** $q \in P_S$ **with** $M(q) = \overline{M}(q) = 0 \ \wedge \ \exists r_k$ **s.t.** $(r_k \in S \wedge y_{r_k}(q) > 0)$**:** Then, if $\exists r_l$ s.t. $r_l \in S \wedge t \in r_l^\bullet$, $t \in \{r_i : r_i \in S\}^\bullet \subseteq (S')^\bullet$. Otherwise, $\exists q' \in (I \cup O \cup P_S) \cap {}^\bullet t$ with $\overline{M}(q') = 0$. Furthermore, since $y_{r_k}(q) > 0$ and, by the sub-case assumption, $\forall r_l \in {}^\bullet t$, $M(r_l) > 0$, it must be that $y_{r_k}(q') > 0$. But then, $t \in \{p \in P_S : M(p) = \overline{M}(p) = 0 \ \wedge \ \exists r_i \text{ s.t. } (r_i \in S \wedge y_{r_i}(p) > 0)\}^\bullet \subseteq (S')^\bullet$. □

Corollary 3 encompasses all the relevant results appear-

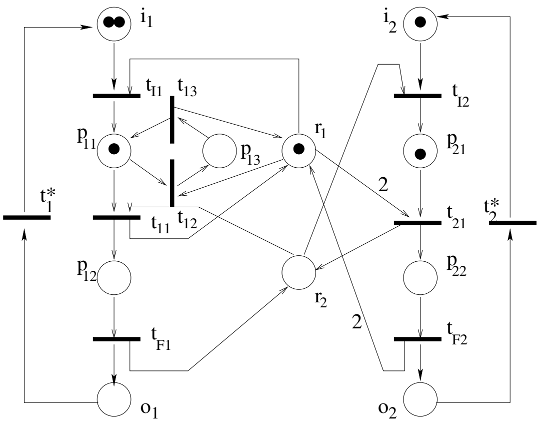

Fig. 2. Example 1 – A case of RAS non-liveness which cannot be attributed to the development of resource-induced deadly marked siphons, due to enabled internal process cycles

ing in [2], [4], [11] and some of the results appearing in [5]. It also subsumes the general siphon-based characterization of liveness for the process-resource nets considered in [14], since that work considers ordinary Petri nets only. The next example demonstrates that for the case of process-resource nets where the process flows can present internal cycles, the structural concept of resource-induced deadly marked siphon might not be sufficient for interpreting the non-liveness of resource allocation in process-resource nets, even under the assumptions of quasi-live and reversible process sub-nets. The works of [15], [16] identify some special structure on the resource allocation requests, that allows the attribution of the net non-liveness to resource-induced deadly marked siphons, even for the case of process-resource nets with cyclic process routes. However, the complete characterization of the dynamics and the liveness-related properties of process-resource nets with cyclic process routes is an issue open to future investigation.

Example 1: Consider an RAS with two resource types, R_1 and R_2, available at 2 and 1 units, respectively, and two process types, J_1 and J_2. Process type J_1 involves three stages, J_{11}, J_{12} and J_{13}, with corresponding resource allocation requests $[1, 0]$, $[0, 1]$ and $[2, 0]$. Process type J_2 involves two stages, J_{21} and J_{22}, with corresponding resource allocation requests $[0, 1]$ and $[2, 0]$. The RAS-modelling process-resource net, $\mathcal{N}$, that expresses also the flow logic defining the possible process transitions among their stages, is depicted in Figure 2. In particular, Figure 2 depicts a marking $M \in R(\mathcal{N}, M_0)$, in which the two active processes are deadlocked (notice that transitions t_{11} and t_{21} are dead in the depicted marking M). Yet, the reader can verify that the corresponding modified marking $\overline{M}$, as well as all the modified markings $\overline{M'} \in \overline{R(\mathcal{N}, \overline{M})}$, contain no resource-induced deadly marked siphon. This results from the fact that the deadlocked process in place p_{11} can circulate freely in the circuit $< p_{11}, t_{12}, p_{13}, t_{13} >$, and therefore, the RAS deadlock of the two processes does not translate to a total deadlock in $\overline{R(\mathcal{N}, \overline{M})}$. □

We conclude this section by formally stating and proving that in the considered class of process-resource nets, liveness and reversibility are equivalent concepts.

Theorem 2: A process-resource net $\mathcal{N} = (P_S \cup I \cup O \cup P_R, T, W, M_0)$ with quasi-live, reversible and acyclic process sub-nets is reversible if and only if it is live.

Proof: The necessity ("only-if") part of this theorem results immediately from Lemma 3, since otherwise $\exists M \in R(\mathcal{N}, M_0)$ s.t. $\exists$ process sub-net $\mathcal{N}_{P_{j^*}}$ with $M(i_{j^*}) + M(o_{j^*}) \neq M_0(i_{j^*})$, and $\overline{M}$ is a total deadlock. In order to establish the sufficiency ("if") part of the theorem, consider a marking $M \in R(\mathcal{N}, M_0)$ s.t. $M \neq M_0$. Then, if for every process sub-net $\mathcal{N}_{P_j}$ it holds that $M(i_j) + M(o_j) = M_0(i_j)$, it should be obvious from the structure of net $\mathcal{N}$ that $M_0 \in R(\mathcal{N}, M)$. Otherwise, using an argument similar to that in the proof of Lemma 3, one can construct a maximal-length firing sequence σ leading to a marking M' s.t. every transition $t \notin (I \cup O)^\bullet$ is disabled in M'. We claim that at M', $\forall$ process sub-net $\mathcal{N}_{P_j}$ it holds that $M'(i_j) + M'(o_j) = M_0(i_j)$, and therefore, $M_0 \in R(\mathcal{N}, M')$, which further implies that $M_0 \in R(\mathcal{N}, M)$. Indeed, by construction, $\overline{M'}$ is a total deadlock of $\mathcal{N}$, and if $\exists$ process sub-net $\mathcal{N}_{P_{j^*}}$ s.t. $M'(i_{j^*}) + M'(o_{j^*}) \neq M_0(i_{j^*})$, Lemma 2 implies that M' contains a resource-induced deadly marked siphon. But then, Theorem 1 implies that $\mathcal{N}$ is not live, which contradicts the working hypothesis. □

IV. Liveness verification and liveness-enforcing supervision for process-resource nets with quasi-live, reversible and acyclic process sub-nets

This section seeks to exploit the siphon-based characterization of non-liveness for process-resource nets with quasi-live, reversible and acyclic process sub-nets, in order to develop computational tools for assessing the liveness of any given instance of the considered PN sub-class, and if necessary, to synthesize a correct liveness-enforcing supervisor (LES). Hence, the first part of the section discusses a sufficiency test for the non-existence of resource-induced deadly marked siphons in the modified reachability space of any process-resource net, that was originally developed in [7]. The considered test extends relevant ideas presented in [4] regarding the detection of the development of empty siphons in ordinary Petri nets, and takes the form of a mathematical programming (MP) formulation, that is polynomially sized, in terms of variables and constraints, with respect to the underlying PN model. As a result, it is very practical from a computational standpoint. The second part of the section describes how the MP-based test discussed in the first part, can be integrated in a synthesis procedure that supports the design of *algebraic* LES for the considered class of process-resource nets.

A mathematical programming-based sufficiency test for the liveness of process-resource nets with quasi-live, reversible and acyclic process sub-nets The starting point for the development of the considered MP-based sufficiency test for the liveness of process-resource nets with quasi-live, reversible and acyclic process sub-nets, is the observation that, given a marked PN $\mathcal{N} = (P, T, W, M_0)$ and a marking $M \in R(\mathcal{N}, M_0)$, the

```
**Input:** A marked PN $\mathcal{N} = (P, T, W, M_0)$ and a marking $M \in R(\mathcal{N}, M_0)$
**Output:** The maximal deadly marked siphon in $M$, $S$
  1. $S := P$;   $\mathcal{N}' := \mathcal{N}$
  2. **while** $\exists\ t \in T$ such that $t$ is fireable in the modified net $\mathcal{N}'$ **do**
    (a) Remove $t$ from $\mathcal{N}'$
    (b) Remove $t^\bullet$ from $\mathcal{N}'$
    (c) $S := S \backslash t^\bullet$
    **endwhile**
  3. **Return** $S$
```

Fig. 3. An algorithm for computing the maximal deadly marked siphon in a given PN marking M

maximal deadly marked siphon S in M can be computed by the algorithm of Figure 3, originally developed in [7]. For the case of *structurally bounded* nets, the algorithm of Figure 3 can be converted to an IP formulation through the use of the *binary indicator* variables v_p, z_t and f_{tp}, respectively defined by the following conditions:

$$\begin{aligned}
v_p = 1 \quad &\Longleftrightarrow \quad \text{place } p \text{ is removed by the algorithm,} \\
&\qquad\quad \forall p \in P \\
z_t = 1 \quad &\Longleftrightarrow \quad \text{transition } t \text{ is removed by the algorithm,} \\
&\qquad\quad \forall t \in T \\
f_{pt} = 1 \quad &\Longleftrightarrow \quad M(p) \geq W(p,t) \ \vee \ v_p = 1, \ \ \forall W(p,t) > 0
\end{aligned}$$

Furthermore, we let $SB(p)$ denote a structural bound for the markings of place $p \in P$. Then, the work of [7] establishes the following theorem:

Theorem 3: Given a marking $M \in R(\mathcal{N}, M_0)$ of a structurally bounded PN $\mathcal{N} = (P, T, W, M_0)$, the maximal deadly marked siphon S contained in M is determined by:

$$S = \{p \in P \mid v_p = 0\} \tag{2}$$

where v_p, $p \in P$, is obtained through the following IP formulation:

$$G(M) = \min \sum_{p \in P} v_p \tag{3}$$

s.t.

$$f_{pt} \geq \frac{M(p) - W(p,t) + 1}{SB(p)}, \quad \forall W(p,t) > 0 \tag{4}$$

$$f_{pt} \geq v_p, \quad \forall W(p,t) > 0 \tag{5}$$

$$z_t \geq \textstyle\sum_{p \in {}^\bullet t} f_{pt} - |{}^\bullet t| + 1, \quad \forall t \in T \tag{6}$$

$$v_p \geq z_t, \quad \forall W(t,p) > 0 \tag{7}$$

$$v_p, z_t, f_{pt} \in \{0,1\}, \ \forall p \in P, \ \forall t \in T \tag{8}$$

□

In order to understand the formulation of Theorem 3, notice that Equation 6 together with Equation 4 imply that all transitions z_t fireable in marking M will have $z_t = 1$. Furthermore, Equation 7 implies that all places $p \in t^\bullet$ for some t with $z_t = 1$ will have $v_p = 1$, which implements Step (2.b) in the algorithm of Figure 3. Similarly, Equation 5 combined with Equation 6 force $z_t = 1$ for all transitions t with $v_p = 1$, $\forall p \in {}^\bullet t$. Finally, the fact that no additional place p (resp., transition t) has $v_p = 1$ (resp., $z_t = 1$), is guaranteed by the specification of the objective function in the above formulation.

In case that net $\mathcal{N}$ is a process-resource net, the formulation of Theorem 3 can be restricted to the computation of the maximal *resource-induced* deadly marked siphon, through the introduction of the following two constraints [7]:

$$\textstyle\sum_{r \in P_R} v_r \leq |P_R| - 1 \tag{9}$$

$$\textstyle\sum_{t \in r^\bullet} f_{rt} - |r^\bullet| + 1 \leq v_r, \quad \forall r \in P_R \tag{10}$$

Specifically, Constraint 9 enforces that the identified siphon S must contain at least one resource place, while Constraint 10 requires that all resource places included in S must be disabling. The resulting necessary and sufficient condition for the non-existence of resource-induced deadly marked siphons in a given marking M of a process-resource net is as follows [7]:

Corollary 4: A given marking M of a process-resource net $\mathcal{N}$ contains no resource-induced deadly marked siphons, if and only if the corresponding formulation of Equations 3–10 is infeasible.

□

The test of Corollary 4 can be extended, in principle, to a test for the non-existence of resource-induced deadly marked siphons over the entire modified reachability space, $\overline{R(\mathcal{N}, M_0)}$, of a process-resource net $\mathcal{N} = (P, T, W, M_0)$, by: (i) substituting marking vector M in the IP formulation of Theorem 3 with the modified marking vector $\overline{M}$; (ii) introducing an additional set of variables, M, representing the net reachable markings; (iii) adding two sets of constraints, the first one linking variables M and $\overline{M}$ according to the logic of Equation 1, and the second one ensuring that the set of feasible values for the variable vector M is equivalent to the PN reachability space $R(\mathcal{N}, M_0)$. Unfortunately, however, any system of linear inequalities exactly characterizing the set $R(\mathcal{N}, M_0)$ is of exponential complexity with respect to the net size [27]. On the other hand, a superset of the reachability space $R(\mathcal{N}, M_0)$ is provided by the system *state equation* [18]:

$$M = M_0 + \Theta \bar{x} \tag{11}$$

$$M \geq 0, \ \bar{x} \in Z^+ \tag{12}$$

The above remarks give rise to a *sufficient* condition for the non-existence of resource-induced deadly marked siphons S in the entire space $\overline{R(\mathcal{N}, M_0)}$ of a given process-resource net $\mathcal{N}$. Furthermore, in the light of Theorem 1, this condition constitutes a *sufficient* condition for liveness of process-resource nets with quasi-live, reversible and acyclic process sub-nets.

Corollary 5: Let $\mathcal{N} = (P, T, W, M_0)$ be a process-resource net with quasi-live, reversible and acyclic process sub-nets. If the mixed integer program defined by (i) Equations 3–10, where vector variable M is replaced by vector variable $\overline{M}$, (ii) Equations 11–12, and (iii) Equation 1, is infeasible, then $\mathcal{N}$ is live.

□

Concluding this paragraph, we notice that for the case of PT-ordinary process-resource nets with quasi-live, reversible and acyclic process sub-nets, a similar but simpler, from a computational standpoint, liveness sufficiency test can be derived, based on the result of Corollary 3. We refer the reader to [4], [6] for a detailed discussion of the corresponding formulation.

Synthesizing correct algebraic LES for process-resource nets with quasi-live, reversible and acyclic process sub-nets Corollary 5 can provide also the basis for the development of a systematic methodology for the design of liveness-enforcing supervisors (LES) for process-resource nets with quasi-live, reversible and acyclic process sub-nets. This is the result of the fundamental stipulation that a LES for any given RAS is *correct* if and only if the controlled system behavior is *live*. Hence, to the extent that Corollary 5 constitutes a liveness criterion, it can provide a LES correctness verification tool, as long as the LES synthesis process is restricted in a way that the controlled system behavior can be modelled by a PN, $\mathcal{N}^c$, that remains in the class of process-resource nets with quasi-live, reversible and acyclic process sub-nets. This can be the case for a large class of RAS LES encountered in the literature, characterized as *algebraic* LES [26], [6]. Essentially, algebraic LES seek to restrict the concurrency supported by the underlying RAS, by setting explicit limits on the number of process instances that can execute simultaneously certain subsets of the RAS process stages. For implementation purposes, this idea is operationalized through the imposition of a set of linear inequalities

$$A \cdot M_S \leq \mathbf{b} \qquad (13)$$

that must always be met by the projection M_S of marking M of the RAS-modelling PN to the subspace defined by its components corresponding to $p \in P_S$. The subset of $R(\mathcal{N}, M_0)$ that is reachable under the observation of Equation 13 constitutes the *LES-admissible* sub-space, $R^{LES}(\mathcal{N}, M_0)$. For a correct algebraic LES, this sub-space must (i) contain the initial marking M_0, and (ii) be strongly connected.

From a representational standpoint, the constraint(s) expressed by Equation 13 can be modelled in the PN-modelling framework, through the theory of *control-place invariants*, presented in [28]. According to [28], each of the inequality constraints

$$\mathbf{a}_{[l,\cdot]} \cdot M_S \leq b_l \qquad (14)$$

can be implemented on the net behavior by superimposing on the original net structure a *control* place w_l, connected to the rest of the network according to the flow matrix

$$\theta_{w_l} = -\mathbf{a}_{[l,\cdot]} \cdot \Theta_S \qquad (15)$$

where Θ_S denotes the flow sub-matrix of the uncontrolled network $\mathcal{N} = (P, T, W, M_0)$ corresponding to places $p \in P_S$. The initial marking of place w_l is set to

$$M_0(w_l) = b_l \qquad (16)$$

and the resulting controller imposes Constraint 14 on the system behavior by establishing the place invariant

$$\mathbf{a}_{[l,\cdot]} \cdot M_S + M(w_l) = b_l \qquad (17)$$

Equation 17, when interpreted in the light of Assumption 4 of Section 2.2, implies that the control places w_l, implementing each of the constraints in the LES-defining Equation 13, essentially play the role of fictitious new resources in the dynamics of the net $\mathcal{N}^c$, that models the controlled system behavior. This observation further implies that the superimposition of an algebraic LES to a process-resource net with quasi-live, reversible and acyclic process sub-nets leads to a controlled net $\mathcal{N}^c$ that falls in to the class of process-resource nets that satisfy Assumptions 2 and 3. However, in order to ensure that the net $\mathcal{N}^c$ satisfies also Assumption 5 with respect to the extended "resource" set $P_R \cup P_W$, some additional restrictions must be imposed on the specification of the LES-defining Equation 13, which will ensure that the constituent process sub-nets remain quasi-live after the introduction of the control places. In that case, it is obvious that the liveness sufficiency condition of Corollary 5, applied on the controlled net $\mathcal{N}^c$, can function as a correctness verification tool for the considered supervisor.

Currently, we lack a complete theory that will address, in the broadest context of process-resource nets, the issue of synthesizing algebraic LES that preserve the quasi-liveness of the constituent RAS processes. An additional limitation of the currently available results is our inability to characterize, for a given process-resource net $\mathcal{N}$, the set of A matrices in Equation 13 that will lead, through Equation 15, to a controlled net $\mathcal{N}^c$ which is *structurally live* with respect to markings $M_0(w_l)$, $l = 1, \ldots, \dim(\mathbf{b})$. Sporadic results, providing algebraic LES classes that preserve the RAS quasi-liveness and are structurally live with respect to markings $M_0(w_l)$, $l = 1, \ldots, \dim(\mathbf{b})$, and that are appropriate for Single-Unit and/or Conjunctive/Disjunctive RAS, can be found in [19], [29], [6], [7].[6] While the development of a complete methodology able to systematically synthesize (algebraic) LES for the class of process-resource nets considered in this work is an important and challenging problem open to further investigation, the results of Sections III and IV can still enable the systematic verification of the correctness of algebraic LES that might have been heuristically developed for any given quasi-live, reversible and acyclic process-resource net. The following example demonstrates this capability.

Example 2: Consider the process-resource net depicted in Figure 4. As it can be seen from the figure, the underlying RAS consists of two processes, J_1 and J_2, and five resource types, $R_1, \ldots, R_5$. Process type J_1 has a flow represented by an acyclic marked graph, and it involves six tasks, $J_{11}, \ldots, J_{16}$, with corresponding resource requirements: $[1, 0, 0, 0, 0]$, $[0, 1, 0, 0, 0]$, $[0, 0, 1, 0, 0]$, $[0, 0, 1, 0, 0]$, $[0, 1, 0, 0, 0]$ and $[0, 0, 0, 0, 1]$. Process type J_2 has a flow represented by an acyclic state machine, and

[6] Also, some of the results presented in [30], [2], [20], [9], [10] can be recast in the same framework.

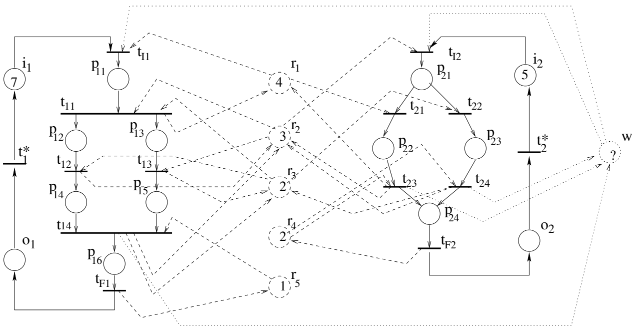

Fig. 4. Example 2 – The considered process-resource net

it involves four stages, $J_{21}, \ldots, J_{24}$, with corresponding resource requirements: $[0,1,0,0,0]$, $[1,1,0,0,0]$, $[0,1,1,0,0]$ and $[0,0,0,1,0]$. A closer inspection of the task/stage resource requirements for these two processes reveals that the only resources that could be entangled in a deadlock are R_1, R_2 and R_3. Therefore, the critical sections for J_1 and J_2 are respectively defined by the stage sets $\{J_{11}, J_{12}, J_{13}, J_{14}, J_{15}\}$ and $\{J_{21}, J_{22}, J_{23}\}$.

Our intention is to develop a LES for this net that will establish the liveness of the controlled net by restricting the number of process instances that can simultaneously execute in their critical sections identified above. Hence, the proposed supervisor constitutes a more refined implementation of the *"process-release"* control scheme, previously proposed in the literature,[7] to the particular process-resource net of Figure 4. The discussion of the previous paragraph suggests that, from an algebraic representational standpoint, the control logic of the considered LES can be expressed by a single linear inequality

$$\mathbf{a} \cdot M_S \leq b \tag{18}$$

where b defines the ceiling on the process concurrency imposed by the considered supervisor, and the elements of the (row) vector $\mathbf{a}$ are provided by a set of p-semiflows characterizing the control flow logic for the various process types in their critical sections. In the PN modeling framework, this LES is superimposed to the original process-resource net of Figure 4 through the introduction of a control place w, connected to the original process-resource net through the flow structure depicted in dotted lines in Figure 4.

Next we seek to determine the maximal marking for w that leads to live behavior for the (controlled) net structure of Figure 4, using the siphon-based liveness analysis developed in this work. For this, we first determine an upper bound to the maximal number of processes that can be executed simultaneously by the considered RAS. The reader can convince herself that, based on the resource capacities and the process flows annotated in Figure 4, an upper bound for the system concurrency w.r.t. job type J_1 (resp., J_2) is 7 (resp., 5) process instances. Then, using the MIP formulation of Corollary 5 in a binary search over the integer set $\{1, \ldots, 12\}$, reveals that the maximal marking for control place w leading to a correct algebraic LES – or equivalently, the maximal number of jobs that can be simultaneously loaded in the system without the possibility of running into any deadlocking problems – is 6. For completeness, we mention that the deadlock marking identified by the computerized solver when the MIP formulation of Corollary 5 was solved with $M_0(w) = 7$, was: $M(i_1) = 1$; $M(p_{11}) = 4$; $M(p_{12}) = M(p_{13}) = 2$; $M(i_2) = 4$; $M(p_{21}) = 1$; $M(r_4) = 2$; $M(r_5) = 1$; and zero for every other place. □

We conclude this section with some remarks on the potential of synthesizing a LES for any given process-resource net, based on the mechanism of process-release control and the structural test for liveness introduced in this section. It is easy to see that, if the process sub-nets of the original process-resource net $\mathcal{N}$ are quasi-live, reversible and acyclic, then the process sub-nets of the controlled net $\mathcal{N}^c$, resulting from the introduction of the control place $\{w\}$, are also reversible, acyclic and quasi-live with respect to the broader "resource" set $P_R \cup \{w\}$, *iff* $M_0(w) \geq 1$. However, $\mathcal{N}^c$ may *not* be structurally live with respect to place w. This effect results from the fact that process nets that contain synchronizing / merging transitions, might need additional control logic to ensure their liveness, even for the execution of a *single* process instance,[8] and exemplifies the extent and nature of the difficulties that must be addressed by any research effort seeking to systematically develop LES appropriate for the considered class of process-resource nets.

V. Conclusions

The work presented in this paper extended the currently existing results regarding the siphon-based characterization of (non-)liveness in sequential RAS. Specifically, it provided a unifying framework for interpreting all the relevant results currently existing in the literature, and even more importantly, it extended the theory applicability to broader RAS classes[9] and it revealed its limitations. It was shown that some key RAS properties that facilitate the interpretation of its non-liveness through the concept of resource-induced deadly marked siphon, are the quasi-liveness, reversibility and acyclicity of its constituent processes.

This finding further suggests that, for RAS with acyclic processes, live behavior can be established through a *two-stage decomposition* procedure, where the first stage seeks to establish inherently consistent – i.e., quasi-live and reversible– process behaviors, and the second stage seeks to develop, if necessary, a control policy, in the form of

[7] c.f., for instance, the policy presented in [2] and some of the policies presented in [20]; however, the results developed in those works will encompass neither the non-ordinary structure of the process-resource net under consideration, nor the complexity of the involved process flows.

[8] Luckily, this effect did not appear in the case of Example 2, in spite of the presence of the synchronizing transition t_{14} in the process type J_1. In fact, the reader should convince herself that, because of the particular structure presented by process subnet between transitions t_{11} and t_{14}, J_1 will be live *iff* it is quasi-live, for *any* given resource availability.

[9] c.f. Example 2 in Section IV, where the addressed process-resource net does not belong to any of the RAS classes that have been studied in the literature.

an (algebraic) LES, that will ensure the RAS liveness in its process enactment phase. In order to support the first stage of this decomposition, more work is necessary towards developing a more profound understanding of the emerging concepts of process quasi-liveness and reversibility, as defined by Assumptions 1, 2 and 5; some preliminary results in this direction can be found in [14]. An issue that needs further investigation in order to support the second stage of the aforementioned decomposition, is the identification – or even better, the development of a mechanism for the automated synthesis – of algebraic LES structures that are guaranteed to maintain the process quasi-liveness of the underlying RAS, and to be structurally live with respect to the marking of the associated control places, in the context of the broader RAS classes considered in this work. Prior experience with the development of similar algebraic LES for the more restricted classes of SU-RAS and CD-RAS can offer useful guidance in this task. Understanding and controlling the dynamics of RAS with internal process cycles is another issue that was shown to lie beyond the boundary of the theory developed in this work, and therefore, it stands open to further investigation. Finally, from an application standpoint, the successful implementation of such a research program will extend our capability towards the effective deployment and (re-)configuration of flexible automation in a broad scope of applications, ranging from automated (e.g., 300mm semiconductor) manufacturing, to driver-less urban mono-rail and railway systems, to web-based workflow management systems.

Acknowledgement

This work was partially supported by the Keck Foundation through the Keck Virtual Factory Lab of the School of Industrial & Systems Engineering at Georgia Institute of Technology.

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*, B. Caillaud, X. Xie, P. Darondeau, and L. Lavagno, Eds., pp. 203–212. Kluwer Academic Publishers, 2002.

[2] J. Ezpeleta, J. M. Colom, and J. Martinez, "A petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. on R&A*, vol. 11, pp. 173–184, 1995.

[3] S. A. Reveliotis, M. A. Lawley, and P. M. Ferreira, "Polynomial complexity deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. on Automatic Control*, vol. 42, pp. 1344–1357, 1997.

[4] F. Chu and X-L. Xie, "Deadlock analysis of petri nets using siphons and mathematical programming," *IEEE Trans. on R&A*, vol. 13, pp. 793–804, 1997.

[5] X. Xie and M. Jeng, "Ercn-merged nets and their analysis using siphons," *IEEE Trans. on R&A*, vol. 13, pp. 692–703, 1999.

[6] J. Park and S. Reveliotis, "Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. on R&A*, vol. 16, pp. 190–195, 2000.

[7] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. on Automatic Control*, vol. 46, pp. 1572–1583, 2001.

[8] S. A. Reveliotis, "Structural analysis of assembly/disassembly resource allocation systems," Tech. Rep., School of Industrial & Systems Eng., Georgia Tech (cond. accepted to IEEE TAC), 2001.

[9] K. Barkaoui and I. Ben Abdallah, "Analysis of a resource allocation problem in fms using structure theory of petri nets," in *Proc. of the 1st Intl Workshop on Manufacturing and Petri Nets*, 1996, pp. 1–15.

[10] K. Barkaoui, A. Chaoui, and B. Zouari, "Supervisory control of discrete event systems based on structure theory of petri nets," in *Proc. of the IEEE Intl Conf. on Systems, Man and Cybernetics*. IEEE, 1997, pp. 3750–3755.

[11] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "A structural approach to the problem of deadlock prevention in processes with resources," in *Proceedings of the 4th Workshop on Discrete Event Systems*. IEE, 1998, pp. 273–278.

[12] F. Tricas, J. M. Colom, and J. Ezpeleta, "A solution to the problem of deadlock in concurrent systems using petri nets and integer linear programming," in *Proceedings of the 11th Eurpoean Simulation Symposium*, 1999, pp. 542–546.

[13] M. P. Fanti, B. Maione, and T. Turchiano, "Comparing digraph and petri net approaches to deadlock avoidance in fms modeling and performance analysis," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 30, pp. 783–798, 2000.

[14] M. Jeng, X. Xie, and M. Y. Peng, "Process nets with resources for manufacturing modeling and their analysis," *IEEE Trans. on Robotics & Automation*, vol. 18, pp. 875–889, 2002.

[15] J. Park, *Structural Analysis and Control of Resource Allocation Systems using Petri nets*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2000.

[16] M. Jeng and X. Xie, "Modeling and analysis of semiconductor manufacturing systems with degraded behaviors using petri nets and siphons," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 576–588, 2001.

[17] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541–580, 1989.

[18] J. Desel and J. Esparza, *Free Choice Petri Nets*, Cambridge Univerrsity Press, 1995.

[19] S. A. Reveliotis and P. M. Ferreira, "Deadlock avoidance policies for automated manufacturing cells," *IEEE Trans. on Robotics & Automation*, vol. 12, pp. 845–857, 1996.

[20] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, "Event-based feedback control for deadlock avoidance in flexible production systems," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 347–363, 1997.

[21] M. Lawley, S. Reveliotis, and P. Ferreira, "The application and evaluation of banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems," *Intl. Jrnl. of Flexible Manufacturing Systems*, vol. 10, pp. 73–100, 1998.

[22] M. A. Lawley, "Deadlock avoidance for production systems with flexible routing," *IEEE Trans. Robotics & Automation*, vol. 15, pp. 497–509, 1999.

[23] W. Van der Aalst, "Structural characterizations of sound workflow nets," Tech. Rep. Computing Science Reports 96/23, Eindhoven University of Technology, 1996.

[24] W. Van der Aalst, "Verification of workflow nets," in *Lecture Notes in Computer Science, Vol. 1248*, P. Azema and G. Balbo, Eds., pp. 407–426. Springer Verlag, 1997.

[25] J. Park and S. A. Reveliotis, "A polynomial-complexity deadlock avoidance policy for sequential resource allocation systems with multiple resource acquisitions and flexible routings," in *Proc. of CDC 2000*. IEEE, 2000.

[26] S. A. Reveliotis, M. A. Lawley, and P. M. Ferreira, "Structural control of large-scale flexibly automated manufacturing systems," in *The Design of Manufacturing Systems*, C. T. Leondes, Ed., pp. 4–1 – 4–34. CRC Press, 2001.

[27] M. Silva, E. Teruel, and J. M. Colom, "Linear algebraic and linear programming techniques for the analysis of place/transition net systems," in *Lecture Notes in Computer Science, Vol. 1491*, W. Reisig and G. Rozenberg, Eds., pp. 309–373. Springer-Verlag, 1998.

[28] J. O. Moody and P. J. Antsaklis, *Supervisory Control of Discrete Event Systems using Petri nets*, Kluwer Academic Pub., Boston, MA, 1998.

[29] M. Lawley, S. Reveliotis, and P. Ferreira, "A correct and scalable deadlock avoidance policy for flexible manufacturing systems," *IEEE Trans. on Robotics & Automation*, vol. 14, pp. 796–809, 1998.

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