Logical Control of Complex Resource Allocation Systems (Extended Abstract)

Spyros Reveliotis *

* *Georgia Institute of Technology, Atlanta, GA, 30332, USA*
(e-mail: spyros@isye.gatech.edu)

A basic positioning of the presented developments. The main problem addressed in this document concerns the coordinated allocation of a finite set of reusable resources to a set of concurrently running processes. These processes execute in a staged manner, and each stage requires a different subset of the system resources for its support. Furthermore, processes will hold upon the resources currently allocated to them until they will secure the necessary resources for their next processing stage.

Such resource allocation dynamics frequently arise in the context of various flexibly automated operations. Some of these operations include: (i) the workflow that takes place in various production shop floors; (ii) the internet-supported platforms that seek to automate certain service operations; (iii) various guidepath-based transport systems, like industrial monorail and urban railway systems; and (iv) the resource allocation that takes place in the context of the contemporary multi-core computer architectures and in quantum computing.

From a theoretical standpoint, the resource allocation problems that are abstracted from the aforementioned applications, correspond to the problem of scheduling a stochastic network with blocking and deadlocking effects. This is an area of the modern scheduling theory with very limited results in the context of the Operations Research (OR) and the Industrial Engineering communities (IE), which are some of the most prominent communities studying resource allocation and scheduling problems. In the author's opinion, this lack of results for the aforementioned class of scheduling problems within the OR and IE communities is due, to a large extent, to the operational complications and the intricacies that arise from the blocking, and especially the deadlocking effects that take place in the underlying class of resource allocation systems (RAS), and prevent a tractable analysis of the corresponding scheduling problems through the modeling and analysis frameworks that have been pursued by those communities.

On the other hand, the control problem of deadlock avoidance and liveness-enforcing supervision for the considered RAS has been systematically investigated within the Discrete Event Systems (DES) community for an extensive time. The main role of this document is to characterize the state of art of these investigations, and to reveal the ability of the corresponding developments to provide effective and very efficient solutions to the supervisory control problem of RAS deadlock avoidance (also known as liveness-enforcing supervision) that is addressed by them.

Furthermore, as indicated in Figure 1, the availability of the aforementioned results can also enable the resolution of the broader scheduling problem for the considered RAS.

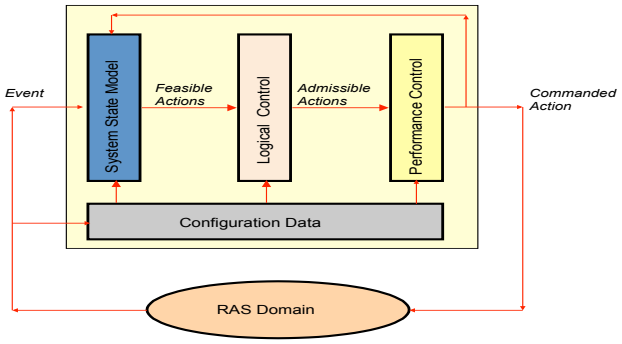

Fig. 1. An event-driven control scheme for the real-time management of the considered RAS. The depicted controller responds to the various events taking place in the "plant" RAS by first updating a state model that defines the feasible behavior generated by this RAS, and subsequently filtering this feasible behavior through a "logical" controller in order to obtain the admissible behavior, i.e., the behavior that is consistent with certain specifications imposed on the RAS operation, including the requirement for deadlock-freedom (or "liveness"). Finally, the admissible behavior is processed through the "performance-oriented" controller in order to select the particular action(s) among the admissible ones that eventually will be commanded upon the RAS.

The controller that is depicted in that figure is based on a two-stage control scheme that decomposes the RAS scheduling problem to (i) the "logical" control problem of deadlock avoidance, and (ii) the subsequent problem of the performance optimization of the logically controlled RAS with respect to (w.r.t.) some typical time-based performance objectives, like the maximization of the long-term throughput or the minimization of various indices of the experienced congestion.

When viewed from a broader methodological standpoint, the presented developments also reveal very vividly the significance of understanding and exploiting, through pertinent representations and customized algorithms, any "special structure" that might be present in the pursued DES applications. At the same time, we believe that many of the key ideas and techniques that have been pursued by the research program presented in this work, hold considerable potential for applicability in other application domains that fall within the scope of DES theory and practice.

Finally, we should also notice that the page limits imposed for this document do not allow the provision of a complete list of references to all the developments that are outlined in this work. But we refer the reader to the recent publication of Reveliotis (2017) for this set of references and for

Table 1. A RAS taxonomy

Based on the structure of Process Sequential Logic	Based on the structure of Resource Requirements
Linear: Each process is defined by a linear sequence of stages	**Single-Unit:** Each stage requires a single unit from a single resource
Disjunctive: A number of alternative process plans encoded by an acyclic digraph	**Single-Type:** Each stage requires an arbitrary number of units, but all from a single resource
Merge-Split: Each process is a fork-join network	**Conjunctive:** Stages require different resources at arbitrary levels
Complex: A combination of the above behaviors	

a much more expansive and systematic exposition of the material that is overviewed in the rest of this document.

The RAS abstraction and the corresponding problem of (maximally permissive) liveness-enforcing supervision. An important step for the systematic study of the resource allocation problems that were outlined in the previous part of this document, was the systematic characterization of these problems through the formal abstraction of the *(sequential) Resource Allocation System (RAS)*. The RAS abstraction is essentially defined by: (i) the "resource" types that are managed by the underlying resource allocation function and their availability; (ii) the sequential logic that defines the various "process" types that are supported by the underlying system; (iii) the resource "requests" that are posed by the different processing stages of the various process types; and (iv) the "protocol" that regulates the allocation of the system resources to the requesting processes. Furthermore, when these RAS models are also used for performance modeling, evaluation and control, the aforementioned information must be further augmented by additional information regarding the timing of the execution of the different processing stages.

Table 1 provides more concrete characterizations of some of the aforementioned informational elements that define the considered RAS. But apart from this expository role, Table 1 has also played a much more fundamental role in the relevant literature, in that the particular RAS structure that is identified in this table defines some of the most prominent RAS classes that have been studied in this literature. Furthermore, it should be clear that the RAS subclasses that are defined in each of the two parts of this table present an increasing conceptual and operational complexity in their underlying dynamics, and therefore, Table 1 has also been a very effective instrument for managing the representational and computational complexity in the development of the relevant theory, and for the systematic classification and understanding of the derived results.

From a more analytical standpoint, the qualitative / behavioral dynamics of the considered RAS w.r.t. their supervisory control problem of deadlock avoidance have been studied primarily through the modeling frameworks of (Extended) Finite State Automata (E-FSA) and Petri nets (PNs). In the PN modeling framework, the PN $\mathcal{N}(\Phi)$ that models any given RAS Φ, essentially consists of a set of subnets that model the sequential logic of the supported process types, and a set of "monitor" places that model the resource allocation function to the contesting processes and the re-usable and invariant nature of the corresponding resource units. In the (E-)FSA modeling framework, the state of the corresponding (E-)FSA $\mathcal{G}(\Phi)$ is a nonnegative integer vector $\mathbf{s}$ that reports the distribution of the active process instances to the various processing stages that are supported by the underlying RAS Φ. State $\mathbf{s}$ evolves through (a) the initiation / "loading" of a new process instance, (b) the advancement of a process instance to a subsequent processing stage, and (c) the termination / "unloading" of a completed process instance. The feasibility of any of these events at a particular state $\mathbf{s}$ is determined by the corresponding resource availability and the imposed resource allocation protocol.

Under, both, the (E-)FSA and the PN representations, the RAS is supposed to start empty, and it is also required to reach this empty state from any state that is reachable by it. Hence, the initial and the only marked state of the (E-)FSA $\mathcal{G}(\Phi)$ is the state $\mathbf{s}_0 = \mathbf{0}$. However, state $\mathbf{s}_0$ might not be reachable in the uncontrolled dynamics of $\mathcal{G}(\Phi)$ due to the formation of deadlocks. Therefore, there is a need for a *"liveness-enforcing" supervisor (LES)* $\mathcal{C}$ that will confine the $\mathcal{G}(\Phi)$ dynamics to a strongly connected component of the corresponding state transition diagram (STD) that contains state $\mathbf{s}_0$. Ideally, we should aim for the *maximally permissive* LES $\mathcal{C}^*(\Phi)$ that will confine $\mathcal{G}(\Phi)$ in its *maximal* strongly connected component that contains state $\mathbf{s}_0$.

Clearly, for any given RAS Φ, the set of the reachable states of $\mathcal{G}(\Phi)$ that are admitted by the corresponding supervisor $\mathcal{C}^*(\Phi)$ is well-defined and unique. In the relevant literature, this set is known as the set of reachable *"safe"* states of Φ, and in the following it will be denoted by S_{rs}. Reachable states of $\mathcal{G}(\Phi)$ that are rejected by $\mathcal{C}^*(\Phi)$ are characterized as *"unsafe"*, and the corresponding set will be denoted by S_{ru}. Hence, a practical realization of the supervisor $\mathcal{C}^*(\Phi)$ must recognize and block any attempted transition of RAS Φ from subspace S_{rs} to subspace S_{ru}.

But the decision problem $\mathbf{s} \in S_{rs}$ is NP-complete, even for the simplest RAS class of Table 1, i.e., the class of Linear, Single-Unit (L-SU) RAS. Hence, unless $P = NP$, the deployment of the target supervisor $\mathcal{C}^*(\Phi)$ will be of super-polynomial complexity w.r.t. the "size" of the underlying RAS Φ, where the latter is defined by the size of any parsimonious representation of its constituent elements. In view of this realization, most of the past work on the considered supervisory control problem of RAS deadlock avoidance has been expended on the specification of sub-optimal (i.e., non-maximally permissive) LES $\mathcal{C}$ that can be designed and implemented with a manageable computational effort; in fact, in certain cases, this effort is polynomially related to the size of the underlying RAS Φ. Furthermore, the literature has also identified some RAS subclasses, of considerable practical interest, for which even the maximally permissive LES $\mathcal{C}^*(\Phi)$ can be deployed with computational cost that is polynomially related to the size of the underlying RAS Φ. But in the rest of this document, we focus on a set of results from a more recent research program that has managed to obtain effective and computationally efficient implementations of the maximally permissive LES $\mathcal{C}^*(\Phi)$ for RAS Φ that come from much more general RAS classes and possess very large structures and state spaces.

Designing and implementing the maximally permissive LES $\mathcal{C}^*(\Phi)$ through classification theory. The new approach for the effective and computationally tractable deployment of the maximally permissive LES, $\mathcal{C}^*(\Phi)$, for a very broad set of the RAS instances Φ falling into the taxonomy of Table 1, is based on the realization that, in the considered application context, the sought supervisor acts as a *"classifier"* for the vectors $\mathbf{s}$ that

constitute the reachable state space, S_r, of the (E-)FSA $\mathcal{G}(\Phi)$; the corresponding classification classes are the two subsets S_{rs} and S_{ru} that partition S_r. Furthermore, the finiteness of S_r, together with the definition of the sets S_{rs} and S_{ru} in the earlier parts of this document, imply that the sets S_{rs} and S_{ru} are effectively computable through standard enumerative algorithms that are provided the classical DES supervisory control (SC) theory. Hence, with these two sets fully available, one can subsequently consider the design of a "classification mechanism" that will be able to resolve efficiently the decision problem "$\mathbf{s} \in S_{rs}$?" (or, equivalently, the complementary problem "$\mathbf{s} \in S_{ru}$?"), for any given $\mathbf{s} \in S_r$, during the real-time operation of the considered RAS Φ.

The effective implementation of the above approach requires (i) the detailed specification of the representations to be employed by the sought classifiers, and (ii) the design of the particular computational methods that will provide the corresponding representations for any given (S_{rs}, S_{ru}) pair. The employed representations must be *"complete"*, i.e., they must be able to provide effective representation of the partition of S_r that is defined by the pair (S_{rs}, S_{ru}), for any instance Φ of the targeted RAS classes. Furthermore, the "on line" classification mechanism that is defined by these representations must involve a minimal computational cost, a requirement that further defines a notion of *"structural minimality"* for these representations. Finally, the necessary algorithms for developing these target representations, for any given RAS instance Φ, must be computationally tractable in spite of the very large size of the underlying state space S_r.

The considered theory has addressed the above requirements by pursuing two primary types of representations for the sought classifiers: (a) *parametric*, and (b) *non-parametric*. One of the simplest parametric representations that can be contemplated for the sought classifiers, is that of a linear system of inequalities that must be satisfied by the states $\mathbf{s} \in S_{rs}$ and violated by each state $\mathbf{s} \in S_{ru}$; such a classifier is characterized as *"linear"* in the relevant literature. In the context of linear classifiers, structural minimality implies the minimization of the number of the employed inequalities. Then, the design of a structurally minimal linear classifier for any pair (S_{rs}, S_{ru}) can be formulated as a mixed integer program (MIP). But this basic formulation is challenged by the very large size of the sets S_{rs} and S_{ru}, that results in extremely large numbers of decision variables and constraints for this formulation. Nevertheless, this complexity problem has been effectively addressed for many RAS classes of Table 1 by taking advantage of a "monotonicity" property that is possessed by the concept of state (un-)safety in these classes, and gives to the corresponding partition (S_{rs}, S_{ru}) the topological structure that is depicted in Figure 2. A first implication of this topological structure is that all the inequalities employed by the sought classifier can be of the "$\leq$" type and with all their coefficients being nonnegative. But even more importantly, the topological structure that is depicted in Figure 2 also implies that, in the design of the sought classifier, one can consider explicitly only the *maximal* elements of the set S_{rs} and the *minimal* elements of the set S_{ru}; for further reference we shall represent these two subsets by $\bar{S}_{rs}$ and $\bar{S}_{ru}$. Typically, the sets $\bar{S}_{rs}$ and $\bar{S}_{ru}$ are smaller than their respective "parent" sets S_{rs} and S_{ru} by many orders of magnitude, and this fact renders much more tractable the aforementioned MIP formulation for the design of the sought classifiers. Finally, additional computational efficiencies in the design of a linear classifier that will represent the partition (S_{rs}, S_{ru}) for a given RAS

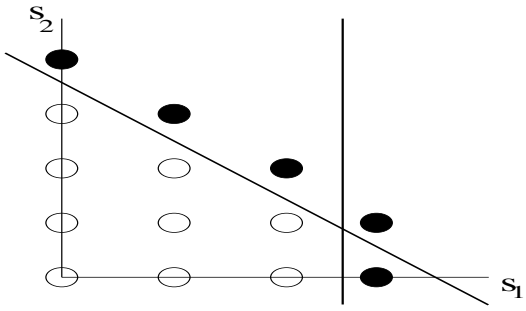

Fig. 2. A drawing characterizing the relative placement of the safe and the (boundary) unsafe states that is implied by the "monotonicity" property of the state safety, for a hypothetical RAS instance Φ with a 2-dim state space. Safe states, identified by while circles, are in the left-lower part of the first orthant of the corresponding 2-dim space, while the unsafe states, depicted by black circles, are in the right-upper part of this orthant.

Φ, can be attained by the realization that each linear inequality employed by the sought classifier acts as a "cover" for the corresponding elements of the set $\bar{S}_{ru}$ that are recognized as "unsafe" by it; this realization enables (i) the adaptation to the considered optimization problem of efficient heuristics that are offered by the existing theory for the classical "minimal set-covering" problem, and (ii) the design of additional customized algorithms for this problem.

On the other hand, linear classifiers are not complete since, in general, the convex hull of the set S_{rs} might contain elements of the set $\bar{S}_{ru}$. Complete classifiers for the considered classification problem can be provided by the broader classification theory that concerns the dichotomy of any given finite vector set to two non-overlapping subsets. Yet, for the considered classification problem, a particular complete classifier can also be obtained by considering the aforementioned "monotonicity" property of state safety and the corresponding topological structure that is depicted in Figure 2. The representation that is employed by this classifier is a *disjunction* of some linear systems of inequalities, and although non-linear in its basic structure and in the employed classification logic, this representation maintains some conceptual affinity to the linear classifier. This affinity has subsequently enabled the extension of many of the aforementioned results on the characterization of the structural complexity and on the computational design of linear classifiers, to this new class of classifiers; these extensions even include the connection of the design of structurally minimal instances from these new classifiers to the minimal set-covering problem.

As for the non-parametric classifiers, they essentially seek to identify and block transitions from S_{rs} to S_{ru} by maintaining an enumeration of the set S^b_{ru} that consists of the *"boundary"* reachable unsafe states, i.e., those reachable unsafe states that can be reached from the set S_{rs} through a single transition. Furthermore, due to the aforementioned "monotonicity" of the (un-)safety concept, the set S^b_{ru} is "right-closed" and it can be effectively represented by an enumeration of its minimal elements; the corresponding set will be denoted by $\bar{S}^b_{ru}$. A major achievement of the relevant theory is the enumeration of the target set $\bar{S}^b_{ru}$ for RAS instances Φ from some major RAS classes of Table 1, without the need for an exhaustive enumeration of the corresponding set S_r. This enumeration relies on a programmatic construction of (i) the minimal deadlocks of the corresponding RAS Φ, and (ii) the minimal deadlock-free unsafe states that can be reached through a pertinent

(but highly non-trivial) "backtracing" process from these minimal deadlocks. Besides the extensive computational efficiency that is established by this procedure, its availability has also enabled the effective deployment of the maximally permissive LES $\mathcal{C}^*(\Phi)$ even for RAS Φ with infinite state spaces, like those RAS that involve resources that can be accessed either in a "reader" or in a "writer" mode, and with the "reader" mode allowing for an arbitrarily large number of readers.

Some additional efforts towards enhancing the representational and computational efficiencies in the construction of a non-parametric representation of the maximally permissive LES $\mathcal{C}^*(\Phi)$ have also sought the employment of BDD-based methods in this construction. And there have also been some attempts to construct incrementally a linear representation of the maximally permissive LES $\mathcal{C}^*(\Phi)$ while foregoing an explicit enumeration of the underlying state space S_r, through an implicit search process that seeks to iteratively identify and block deadlock states that remain potentially reachable in the state space of the underlying RAS Φ. The search for these deadlock states is performed through the development and solution of a mathematical programming (typically a MIP, but in a certain case even a SAT) formulation. But as already mentioned, there are many RAS Φ that will not admit a linear representation for the target LES $\mathcal{C}^*(\Phi)$. And even in the case where such a linear representation of the LES $\mathcal{C}^*(\Phi)$ is possible, the last set of methods mentioned above seem to be challenged by an inability to attain structural minimality for the constructed LES.

Finally, it is also worth-noticing that the aforementioned theory for representing the sought supervisors for the various RAS classes of Table 1 as a classifier, recently has been extended through the concept of the *"maximal" linear classifier*. A maximal linear classifier is a sub-optimal LES C for those RAS Φ that will not admit a linear representation for the corresponding maximally permissive LES $\mathcal{C}^*(\Phi)$, which possesses a linear structure and attains a notion of "maximal permissiveness" within the scope of linear classifiers. The corresponding theory also provides complete algorithms for the computation of such maximal linear classifiers with a minimized structure, that build upon the previously developed theory and methods for the computation of a (structurally minimal) linear representation for the maximally permissive LES $\mathcal{C}^*(\Phi)$, whenever such a representation is possible.

Ongoing and future work. All the above discussion has revealed the richness, and also the extensive potential for practical applicability, of the existing results on the liveness-enforcing supervision for the various RAS classes of Table 1. A remaining important issue is the complementation of these results with an additional set of results that will address the "performance-control" part of Figure 1. Some important developments along this direction are as follows:

For RAS exhibiting a stationary nature for their underlying timed dynamics, the scheduling problem of the logically controlled RAS can be modeled as an average-reward Markov decision process (AR-MDP), and this model extends even to non-Markovian timed dynamics through the method of stages. Furthermore, the confinement of the RAS dynamics into a single strongly connected component of the underlying state space by means of the employed LES $\mathcal{C}$, places the aforementioned MDP into the class of "communicating" MDPs, and renders it solvable through extensively studied and well understood algorithms.

But the effective application of these classical MDP algorithms is practically limited by the very large size of the underlying state spaces. In fact, the size of these state spaces, when combined with their discrete nature, renders prohibitive not only the computation, but even the mere enumeration of an optimal scheduling policy for the considered RAS.

Two possible ways to circumvent this high representational and computational complexity are: (i) the employment of some parameterized policy spaces that admit a more parsimonious representation than the standard tabular representation of the optimal DAP, and the solution of the corresponding scheduling problem over these parameterized policy spaces; and (ii) the "on-line" determination of an optimized scheduling policy for the underlying RAS through the exploitation of the information that is provided in the optimal solution of a simplified version of the original scheduling problem, which is typically known as the corresponding "fluid relaxation". Preliminary investigations with both of these classes of methods have shown quite promising results.

Finally, for the scheduling of non-stationary RAS, one can consider the adaptation of model predictive control (MPC) type of methods, where the "stability" concept in the classical MPC theory is substituted by the notion of the RAS liveness. Results along these lines have recently been developed in the context of the effective and efficient scheduling of some guidepath-based traffic systems that are encountered in modern production and distribution environments and in quantum computing.

REFERENCES

Reveliotis, S. (2017). Logical Control of Complex Resource Allocation Systems. *NOW Series on Foundations and Trends in Systems and Control*, 4, 1–224.