Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-7 (2015) 050–057

Real-Time Management of Complex Resource Allocation Systems: Necessity, Achievements and Further Challenges [⋆,⋆⋆]

Spyros Reveliotis [*]

[*] *School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA (e-mail: spyros@isye.gatech.edu).*

Abstract: Many contemporary applications, ranging from flexibly automated production systems, to automated material handling and intelligent transportation systems, to internet-based workflow management systems, and more recently, to the massively parallelized software systems that emerge in the context of the novel multi-core computing architectures, can be perceived as a set of finite resources that support a number of concurrently running processes; these processes execute in a staged manner and vie for the allocation of various subsets of the system resources. To effectively support and manage the extensive levels of concurrency and operational flexibility that are contemplated for these environments, and the ensuing complexity, there is a substantial need for formal models and tools that will enable the modeling, analysis and eventually the control of the aforementioned resource allocation function so that the resulting dynamics are, both, behaviorally correct and operationally efficient. This write-up overviews a research program that seeks to address the aforementioned need by using the unifying abstraction of the resource allocation system (RAS) and supporting modeling frameworks, like automata, Petri nets, and Markov reward and decision processes, borrowed from the area of Discrete Event Systems (DES) theory. The presented results take advantage of the special structure that exists in the considered RAS classes, and are further characterized by, both, their analytical rigor and computational tractability. The write-up also highlights the further challenges that must be addressed for the successful completion and promotion of the pursued framework.

Keywords: Resource Allocation Systems, Discrete Event Systems, Supervisory Control, Deadlock Avoidance, (Stochastic) Scheduling

1. INTRODUCTION

The efficient and expedient allocation of a finite set of resources to a number of contesting processes is a ubiquitous problem, arising in various operational settings of our contemporary technological civilization. Indeed, cost-effectiveness and responsiveness are predominant concepts in modern corporate strategy and typical requirements for many everyday functions. In the context of the resource allocation functions considered in this work, cost-effectiveness is based on the ability to "make the most" – i.e., maintain a high utilization – of the engaged resources. On the other hand, the posed requirement for high responsiveness traditionally has implied the ability to fill an arising demand or to support an emerging service need in a timely manner. But in the current competitive markets, responsiveness also implies the ability to provide a broad range of diversified products and services, each of them appealing to a different market segment. And this trend for extensive diversification has more lately evolved to the mass-customization practices that have been adopted by certain industries, like the automotive and computer manufacturers, and are contemplated by many more. The effective support of all the aforestated requirements necessitates the further deployment, at the operational level, of high levels of concurrency and flexibility; i.e., a need to support the simultaneous execution, on the same set of resources, of a broad set of diversified workflows, while each of these workflows takes place at a low or moderate rate, and evolves continually into new operational patterns.

A concrete manifestation of the aforementioned trend is provided by the proverbial concept of the "flexible manufacturing system" (FMS), that has been extensively discussed in the context of various industrial settings for many years. The prototypical FMS consists of a set of numerically controlled workstations interconnected by an automated material handling system, and with the workflow taking place in the entire facility being integrated and coordinated by a computerized controller. More recently, this trend has been extended to the service sector through the concept of the "workflow management system", i.e., a computerized tool that supports the definition, the enactment and the coordination of the execution of business

[⋆] This work has been partially supported by NSF grant ECCS-1405156.

[⋆⋆] For its most part, this write-up constitutes an extended abstract of the material that is presented in Reveliotis (2015). We refer to that work for a more systematic and extensive coverage of the presented material, and for the corresponding bibliography.

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.472

workflows, by monitoring their progress and assigning the necessary resources to them; these resources can range from data files, to supporting processing software, to printing and communication means, and even the humans that might be necessary for the support and the authorization of certain steps. It is currently believed that various routine transactions in the banking sector, in the supply chain management and logistics services, in insurance claim processing, and even in the broader health-care sector, can be rendered more responsive and efficient through their mechanization by the successful deployment of a workflow management system.

In the transportation sector, the aforementioned resource allocation paradigm manifests itself in any set-up where a set of concurrently traveling vehicles have to negotiate the necessary traveling space in the context of some "zoning scheme" that ensures collision-free operation. Trains being successively allocated a sequence of segments of the underlying railway network is a typical example of such a "zone"-based operation. Also, in the industrial sector, similar zoning schemes have been implemented in the operation of unit-load automated guided vehicle (AGV) and monorail-based material-handling systems, and in the operation of the hoist and the crane systems that support the material-handling operations at many ports and freight terminals.

Finally, another domain where the aforementioned resource allocation functions (and problems) are very prominent, is in the software and the computational platforms that control the aforementioned operations as well as any other operation that is supported by our modern technological civilization. Indeed, since its early days, our modern computing technology has employed extensive levels of (actual or virtual) concurrency, where a number of software threads run in parallel, each of them tasked with a particular role and function. These threads need to share the limited resources of the underlying computer platform (CPUs, registers, I/O devices, files, etc.) in a way that provides to each of these processes exclusive access and engagement of these resources; the corresponding coordination is attained through the use of a set of tokens, that are typically known as "mutually exclusive (mutex) locks" or "semaphores", and constitute essentially a "pass" for accessing the corresponding resource.

The above discussion regarding the increased levels of efficiency / cost-effectiveness, responsiveness, concurrency and flexibility that are requested for many contemporary operations, and the accompanying examples, also render pretty clear that these requirements lead to operations that are characterized by a high level of operational complexity. And this complexity translates to some challenging scheduling problems for the underlying resource allocation functions. In many cases, including all of the aforementioned examples, things are further complicated by the extensive levels of automation and autonomy that is requested by the considered operations. The need for automation can arise from technological and/or feasibility considerations (as in the case of the multithreaded software mentioned above, and in the operations taking place in the modern semiconductor fabs that must be isolated from the polluting effect of the human element), or from considerations concerning the operational and financial efficiency of the underlying operation (as in the aforementioned workflow management systems and the driverless transportation systems). In either case, the removal of the human element from the underlying processes implies that the deployed controllers, and especially the resource allocation functions involved, must be not only efficient but also correct and robust to logical problems and errors that, in more traditional settings, have been addressed by human intervention and improvisation. A typical such logical problem in the context of the considered resource allocation functions is the formation of (partial) deadlock, i.e., situations where a set of the concurrently executing processes are entangled in a circular waiting pattern, each of them waiting for some of the other processes to release resources that are necessary for its advancement. Hence, any such deadlock is a pernicious situation that stalls the further advancement of the processes involved and drives to zero the utilization of all the resources that have been allocated to these processes. At the same time, it should be obvious that deadlock is a natural consequence of the concurrency and the flexibility, and, finally, the arbitrary structure of the corresponding resource allocation function that is implied by the first two requirements; therefore, it constitutes a ubiquitous problem for the operational environments discussed in the previous paragraphs.

The bottom line of all the above discussion is that, in the context of the considered automated operations, the underlying resource allocation functions must be controlled for operational efficiency, cost effectiveness and responsiveness, and also for correctness and robustness to certain problems of a more qualitative or "logical" nature, like the formation of partial deadlock. In the rest of this write-up we present the results of a research program that has sought to provide a systematic and rigorous solution to this challenging resource allocation problem by employing and extending results from modern control theory.

2. RESOURCE ALLOCATION SYSTEMS AND THE PROPOSED CONTROL FRAMEWORK

The presented research program has sought to address the control problem that was outlined in the introductory section, in a systematic and rigorous manner, by (i) abstracting the considered operations through the notion of a (sequential) Resource Allocation System (RAS), and (ii) employing and extending results coming from the controls area of Discrete Event Systems (DES). In this section, first we introduce the formal notion of the sequential RAS, and subsequently we outline the DES-based control framework that has been proposed for these RAS. We also present a RAS taxonomy that has been instrumental in the investigation of the relevant control problems. These problems, themselves, and the currently available results for them, as well as the remaining open challenges, will be addressed in subsequent sections.

Sequential Resource Allocation Systems. A sequential RAS is formally defined by a quintuple $\Phi = \langle \mathcal{R}, C, \mathcal{P}, \mathcal{A}, \mathcal{D} \rangle$, where: (i) $\mathcal{R} = \{R_1, \ldots, R_m\}$ is the set of the system *resource types*. (ii) $C : \mathcal{R} \rightarrow \mathbb{Z}^+$ – the set of strictly positive integers – is the system *capacity* function, characterizing the number of identical units from each resource type available in the system. Resources are assumed to be