Towards Dynamic Resource Management in Business Processes*

Sven Ihde[1], Luise Pufahl[1], Asvin Goel[2], and Mathias Weske[1]

[1] Hasso Plattner Institute, University of Potsdam, 14482 Potsdam, Germany
{sven.ihde,luise.pufahl,mathias.weske}@hpi.de
[2] Kühne Logistics University, Hamburg, Germany
asvin.goel@the-klu.org

Abstract. Resource management is an important part of modern business processes. Although some extensions for handling resource allocation within the modeling standard *Business Process Model and Notation* have been proposed before, we argue that these are limited for the dynamic management of resources in business processes, where an efficient allocation of resources to tasks is of crucial importance for achieving overall efficiency. In this paper we show challenges of resource management based on a real-world example, and why current approaches are not able to fully handle them. We propose to extend existing business process management systems by a so-called *Resource Manager* with which powerful optimization algorithms for resource allocations can be used and discuss its main functionality and remaining challenges.

Keywords: Resource Allocation · Resource Optimization · Business Process Modeling · Business Process Architecture.

1 Introduction

Business processes are indispensable for organizations to achieve certain goals, such as selling products or hiring employees. Especially for a business, its processes increase in amount and complexity alongside with its growth [5]. This raises the need for concepts, methods, and techniques to represent business processes in such a way, that the process experts can analyze, comprehend, and improve existing business processes [15]. The main artifact is thereby a process model representing visually a certain business process. The de facto standard *Business Process Model and Notation* (BPMN) [10] is commonly used as process modeling notation. The resulting model can also be used to automatically execute a process with business process management systems (BPMSs) [5, 15].

For enacting business processes, resources (e.g. employees, organizations, hardware, software, etc.) are indispensable [15]. Given that many resources are cost-intensive and limited, a main challenge for efficient process execution is to

* The research leading to these results has been partly funded by the BMWi under grant agreement 01MD18012C, Project SMile. http://smile-project.de

find a perfect balance between having enough resources available at all times in order to execute processes without delays, and not having too many resources available which are under-utilized [1]. For the case of human resources, there exist so called resource allocation patterns [12], with are used to pre-define at design time to allocate resources at runtime.

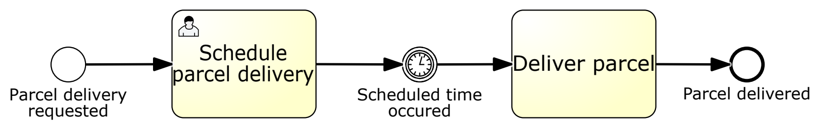

Fig. 1. Simplified model of a parcel delivery process

Rule-based allocations of resources to tasks cannot leverage the full potential of an efficient resource management. The example in Fig. 1 shows a very simple process model of a parcel delivery process where each parcel has its own process instance. For each parcel, the delivery is scheduled and when the scheduled time is reached, the parcel is delivered to the customer. Clearly, resources, i.e., delivery vehicles, are needed in order to actually conduct the delivery. In order to efficiently utilize these vehicles, the scheduling of the individual parcels must be conducted in such a way that cost-efficient routes for the vehicles can be found, allowing to deliver all parcels at their scheduled times. Thus, the scheduling represented as a simple user task (cf. Fig. 1) is a complex task, where all suitable vehicles and all other delivery requests should be considered. In our example, we assume that this task is done manually by a dispatcher, who has knowledge about all resources available and all delivery requests. However, the task could also be executed by an optimization engine using algorithms for solving a so-called *vehicle routing problem* [14], a combinatorial optimization problem known to be $\mathcal{NP}$-hard, and thus, being extremely difficult to solve.

Similar combinatorial optimization problems, in which rule-based allocations of tasks to resources can lead to significant efficiency problems, can also be found in many other business processes. Therefore, an innovative approach that enables current business processes to be extended by the means to dynamically manage resources is needed. We propose an extension of existing Business Process Enactment Systems by introducing a new component, which is responsible for resource allocations – the `Resource Manager`. The `Resource Manager` is supposed to accumulate and congregate the knowledge of the different resources, as well as having access to currently running processes, resulting in a highly adaptive resource allocation.

In the next section 2, existing approaches concerning resource allocation are discussed. In section 3 we introduce the `Resource Manager` and what requirements it has to fulfill. On top of that, we will explain how our approach can fill the shortcoming of the current approaches in accordance to the given example.

Lastly, in section 4 we will give an outlook on how we will work to realize our proposal as well as validate its functionality.

2 Related Work

Next to the control-flow and data-flow perspective, the resource-perspective is essential for business processes describing how to allocate required resources to a process task for its successful execution [15]. So called Resource Allocation Patterns [12] have been introduced to manage (mostly) human resources. Further it was shown (see [2, 6, 16]) that these resource allocation patterns could be used during process modeling by adapting the existing modeling notation BPMN [10]. These resource-aware Process Models (see [7, 13]), where then analyzed by Cabanillas et al. [3, 4]. They stated that these models tend to be too complex for a process modeler to use or only offer a limited flexibility of resource allocation rules. When looking at non-human resources, there exist multiple approaches (see [9, 11]) that try to efficiently combine process instances for certain activities (i.e., batching) in order to optimize the resource utilization. However, these approaches only support a fixed set of allocation algorithms or only resources of one type, which, especially with upcoming trends like Internet of Things (see [8]) and digitization, is not enough. Given that the problem of allocating resources to tasks is a complex problem that can be tackled in many different ways (e.g. manually or by use of optimization algorithms), we can observe that there is a need to explicitly model an entity for making resource allocation decisions, allowing a process designer to focus on the process model, whereas the resource allocation decisions can be optimized by a dedicated engine having access to all relevant information about resources and requests for these resources.

3 Resource Manager

In this section, we introduce a new approach that extends the current business process management system by adding one further component (see Fig. 2). Traditionally, a BPMS consists of two main components: the `Process Modeling` and the `Process Engine` component. With the `Modeling` component, process models can be created by a process designer and stored into a `Process Model Repository`. The current versions of the models in the `Repository` are then accessed by the `Engine` to execute them. If the process contains service tasks, the `Engine` calls other applications – in the case of a user task, the `Engine` provides the tasks to the process participants over an user interface via the `Environment`.

The new component is an additional service for business processes, which allows an optimized resource allocation. It is independent from the individual process execution, but has on the other hand knowledge on the existing resources, their characteristics (static data), their availability (dynamic data), and on the other hand, access to the currently executing process cases (i.e. process instances). Our extension is built up similarly to the traditional BPMS and consists of three parts: `Resource Modeling`, `Resources`, and `Resource Manager`.

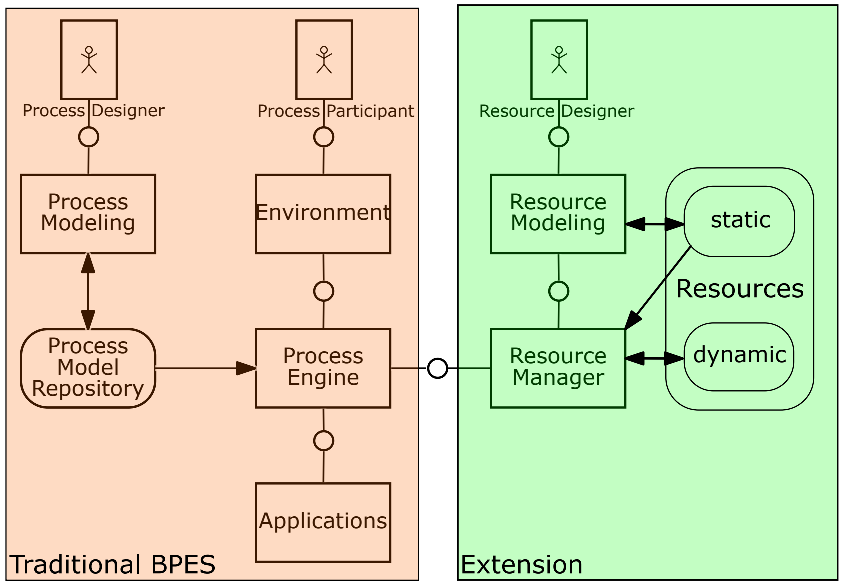

Fig. 2. FMC model of our proposal for the extended Business Process Management System

Resource Modeling. Similar to the `Process Modeling`, this component is used by the `Resource Designer` to model the static part of resources according to a notation, that can be translated and executed by the `Resource Manager`. The `Resource Modeling` is responsible for defining the static data required to determine whether a resource may be allocated to a request or not, as well as the data required to assess and evaluate resource allocations. For example, the data may include the type of the resource, the hourly costs, capacity, etc.

Resources. The `Resources` contain the full information on the static information of resources that are supposed to be modeled. During runtime the `Resources` additionally contain dynamic data, which is updated by the `Resource Manager` only. In this way the `Resources` part contains the exact representation of the existing resources. This means if a resource can not be found in the `Resources` it does not exist and if we find two resources of the same type, we have exactly two resources. Additionally we propose to give read-only access to the `Resources` to the `Process Designer` during `Process Modelling` in order to make sure if the process is going to use resources, it is done correctly. Thanks to that, in contrast to traditional process designing, the `Process Designer` does not need to specify the concrete resource for an activity but can instead ask the `Resource Manager` to allocate the needed resource at run-time. This means that there is the need of a change in the `Process Modeling`, whenever a resource is a part of the process (see Fig. 3).

Resource Manager. The `Resource Manager` is a component maintaining information about the current state (dynamic data) of all resources. Whenever a resource is requested by a process instance, it collects all relevant data concerning the resource request so that allocation decisions can be made. The actual allocation decisions can then either be performed via an appropriate user interface by a human decision maker or via a suitable interface by an optimization engine. Whenever a resource is allocated to one or multiple requests, the state of the resource is updated and the respective process instances are informed about the allocation. Whenever a requested resource is no longer needed, the `Resource Manager` is informed by the process instance so that the state of the resource can be updated accordingly.

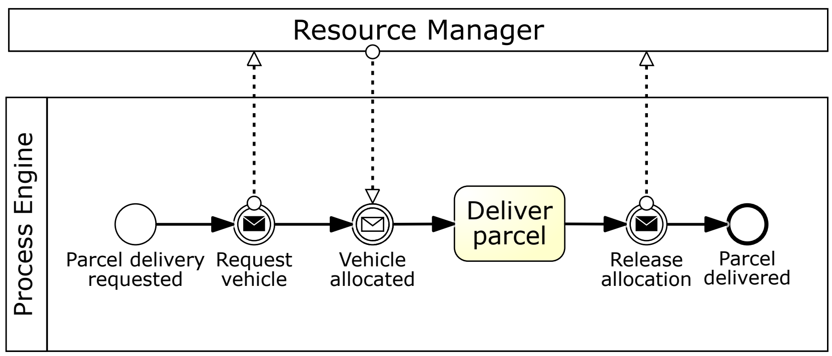

Fig. 3. Parcel delivery process with resource manager

Fig. 3 shows an adaptation of the parcel deliver process in which the communication between the process model and the `Resource Manager` is explicitly modelled. In comparison to the process shown in Fig. 1, the scheduling task is no longer executed within the process instance itself. Thus, instead of having the planning tasks distributed over all individual process instances, the scheduling task, which should consider information about all delivery requests and vehicles, can be executed where all of this information is available. As the `Resource Manager` has to be called from inside the `Engine` that executes the business process, there has to be a close connection between these two components. Another idea would be to fully incorporate the `Resource Manager` into the `Engine`. However, this will lead to problems as the `Resource Manager` is supposed to manage the resources for multiple processes (instances) as those different execution might not necessarily be on the same `Engine`. That is why we need to have the `Resource Manager` as a stand-alone component. Furthermore, the `Resource Manager` is expected to be the expert concerning all resources it is in charge of. This means it further encapsulates the resource specific knowledge as not all processes need the complete information or on different detail levels.

4 Future Work

In this work we introduced a concept to extend the architecture of BPMS in such a way that an effective linking between tasks and resources is possible to cope with challenges from multiple and even freshly emerging fields (e.g. IoT, parcel delivery, cloud etc.). However, so far we only theoretically showed the shortcomings of existing approaches and how a component managing resources as a service could bridge these gaps. In order to evaluate our approach, we are going to implement a proof of concept. We have identified three main challenges we need to accomplish. Firstly, as mentioned during the explanation of the `Resource Modeling`, the challenge is to define a standard notation for resource models. After we have resources defined in such a way processes can use them, we can create resource-aware business processes. For this we need to adapt the `Process Modeling` part, so that a process now can access resources when needed. This means we will have to introduce new modeling concepts concerning resources. Lastly we are going to implement a `Resource Manager` that at runtime uses its expertise to enable dynamic resource allocation. As the `Resource Manager` is supposed to be the expert for many different allocation problems, we need to define the communication between the `Engine` and `Resource Manager`. For this we are going to define an interface for each allocation problem the `Resource Manager` can solve, where the `Engine` will send its constraints and expectation on the result, which are used to come up with an efficient allocation. We are going to evaluate the implementation on a real-world example in the context of a logistics project (SMile Project). Depending on these results we can then further refine our approach.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (Apr 2010). https://doi.org/10.1145/1721654.1721672
2. Awad, A., Grosskopf, A., Meyer, A., Weske, M.: Enabling resource assignment constraints in BPMN. Tech. rep., BPT Technical Report 04, Hasso Plattner Institute, Potsdam (2009)
3. Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., Ruiz-Cortés, A.: Ralph: a graphical notation for resource assignments in business processes. In: CAiSE. pp. 53–68. Springer (2015)
4. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Ral: A high-level user-oriented resource assignment language for business processes. In: CAiSE. pp. 50–61. Springer (2011)
5. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018). https://doi.org/10.1007/978-3-662-56509-4
6. Großkopf, A.: An extended resource information layer for BPMN. Master's thesis, Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam (2007)
7. Künzle, V., Weber, B., Reichert, M.: Object-aware business processes: Fundamental requirements and their support in existing approaches. International Journal of Information System Modeling and Design (IJISMD) **2**(2), 19–46 (2011)

8. Meyer, S., Ruppen, A., Magerkurth, C.: Internet of things-aware process modeling: integrating IoT devices as business process resources. In: CAiSE. pp. 84–98. Springer (2013)
9. Natschläger, C., Bögl, A., Geist, V.: Optimizing resource utilization by combining running business process instances. In: Service-Oriented Computing-ICSOC 2014 Workshops. pp. 120–126. Springer (2015)
10. OMG: Notation BPMN version 2.0. OMG Specification, Object Management Group (2011)
11. Pufahl, L., Weske, M.: Batch activities in process modeling and execution. In: International Conference on Service-Oriented Computing. pp. 283–297. Springer (2013)
12. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: CAiSE. pp. 216–232. Springer (2005)
13. Saidani, O., Nurcan, S.: Towards context aware business process modelling. In: BPMDS'07, CAiSE. vol. 7, p. 1 (2007)
14. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications. No. 18 in MOS-SIAM Series on Optimization, SIAM (2014)
15. Weske, M.: Business Process Management - Concepts, Languages, Architectures, 2nd Edition. Springer (2012). https://doi.org/10.1007/978-3-642-28616-2
16. Wolter, C., Schaad, A.: Modeling of task-based authorization constraints in BPMN. In: CAiSE. pp. 64–79. Springer (2007)