



Integrating Robotic Process Automation into Business Process Management

Maximilian König^(✉), Leon Bein^(✉), Adriatik Nikaj, and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{Maximilian.Koenig, Leon.Bein}@student.hpi.de,
{Adriatik.Nikaj, Mathias.Weske}@hpi.de

Abstract. As of today, robotic process automation (RPA) is a prominent process automation technology, which uses software to replace humans at operating graphical user interfaces. However, RPA is limited in scope and, in order for it to be established successfully, its environment must meet many requirements. The more mature research field of business process management (BPM) has the potential to provide the environment for RPA to thrive. We present an approach for embedding RPA into BPM in order to link their technologies and combine their systematic methods. The approach allows RPA to synergize with capabilities and insights provided by BPM.

Keywords: Robotic process automation · Business process management · RPA architectures

1 Introduction

Robotic process automation (RPA) is an emerging technology to automate business processes that are driven by user interaction with software systems. It is characterized as generic term for software that mimics human interaction with graphical application interfaces [1]. Thus, human resources are replaced by software robots, which results in decreasing costs and increasing efficiency and consistency [5]. The emergence of RPA is an important development in process automation, labeled as “fastest-growing software subsegment” in 2018 by the IT market research company Gartner¹.

However, RPA is limited in that many techniques required to successfully implement it lay outside its scope. This includes gathering the necessary information for automation enactment, dealing with exceptions during the execution of automated processes, and managing process automation on an organizational level.

Existing research suggests to solve these problems by combining RPA with business process management (BPM). More specific, most works propose integrating RPA with BPM. RPA is considered more successful, or even only successful, when combined with BPM [5, 6, 8, 9]. While Kirchmer et al. [5] already

¹ <https://www.gartner.com/en/newsroom/press-releases/2019-06-24-gartner-says-worldwide-robotic-process-automation-sof>.

present a so-called value-driven robotic process automation approach, including formal methods for RPA enactment and suggesting to integrate them into BPM, no concrete solution for the integration is described.

In this paper, we propose an integration solution, from a software architecture as well as a methodology perspective, to position RPA into BPM. To evaluate our approach, we implemented a prototypical software solution and applied our approach to a use case scenario.

The remainder of this paper is structured as follows: Sect. 2 describes the fundamentals of RPA and BPM. Following, Sect. 3 lists the limitations of RPA to motivate the need for putting it into a larger context and allow for a discussion of our work. In Sect. 4, we explore the existing work that suggests an RPA-to-BPM integration, and examine the proposed approaches. Section 5 and Sect. 6 present our main contribution: A concrete integration solution, consisting of software architectural and methodological means to implement RPA processes in a BPM context. Section 7 evaluates the technological feasibility, applies our solution to a use case, and discusses general findings and shortcomings. In Sect. 8, the main results are summarized and future work is investigated.

2 Preliminaries

This section outlines our understanding and assumptions about BPM and RPA. A general architecture of the underlying systems of BPM and RPA is detailed as it serves the upcoming sections.

2.1 Business Process Management

According to the definition provided by Weske, “BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes” [10]. It is a mature research area that encompasses rich knowledge from academia and industry as well.

The methods needed for conducting a successful BPM project can be structured into the BPM lifecycle. The lifecycle provides an iterative methodology for the enactment of BPM on the level of business processes. While the exact phases differ from source to source, the included activities and their order stay the same. In this paper, we will follow the definition by Weske [10], which

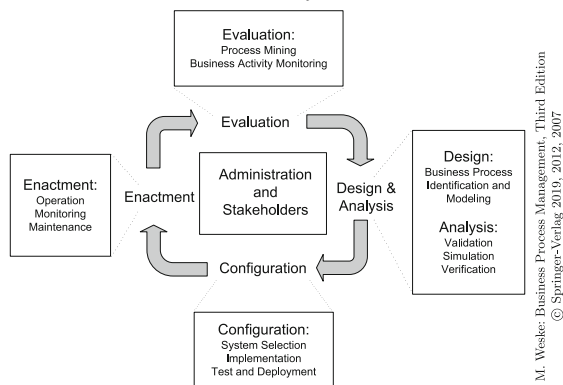


Fig. 1.5. Business process lifecycle

Fig. 1. BPM lifecycle taken from [10]

is depicted in Fig. 1. It is structured as follows: The entry point to the cycle is the *design and analysis* phase, where the business processes are identified and provided with a formal representation. Newly created models and models from past iterations are verified and validated against current process requirements. In the *configuration* phase, the systems to use are selected, and the business processes identified before are implemented, tested, and deployed. During the *enactment* phase, the processes are operated, and the process execution is monitored and maintained. The resulting execution data is processed by the techniques of the *evaluation* phase, for example process mining. Using the knowledge gained from one iteration, the next iteration can be started by redesigning the business processes.

Business process management is realized by business process management systems (BPMS). Their generic architecture is shown in Fig. 2.

A BPMS consists of a collection of tools that allow the model-driven enactment and operation of business processes. It includes a business process modeler, which allows a process designer to model and configure business processes and to deploy them, a process engine, which executes the process models with the help of external applications, and a graphical user interface, which allows process participants to operate and monitor this execution. Analysis and evaluation tools like process miners and other peripheral tools are often shipped with BPM systems and are an important part of BPM tooling. A BPMS allows for automation of business processes by delegating the execution of specific tasks to software with APIs. It further enables process orchestration, resource management, process monitoring and process analysis.

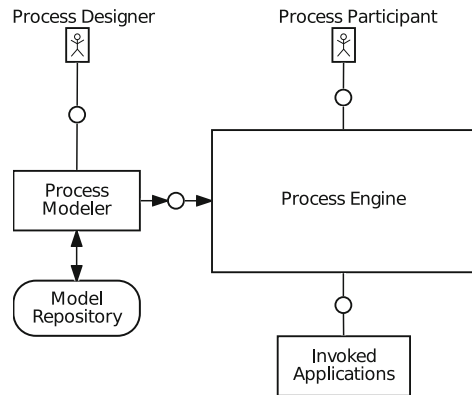


Fig. 2. BPMS architecture adapted from [10]

2.2 Robotic Process Automation

Defined by van der Aalst, “RPA is an umbrella term for tools that operate on the user interface of other computer systems in the way a human would do” [1]. RPA is an upcoming “hot topic” for research and companies. Many projects and research works have been realized in this context within the last few years. RPA aims at automating business processes that consist of human interaction with software, such as transferring data from an ERP system to a web application form. Thereby, human involvement is reduced to starting and supervising the automated processes, a role which will be called *operator* in the course of this paper. Processes automated with RPA will be referred to as *RPA processes*. As there is no standardized formalization yet, their model representations depend on the RPA provider.

RPA is enacted by robotic process automation systems (RPAS). The basic structure of an RPAS, described in Fig. 3, looks similar to that of a BPMS:

The modeler allows an RPA process designer to create RPA process models and to deploy them to a model repository. The controller holds the repository and orchestrates running robots as resources to execute RPA process instances. It provides an interface to the operator to start and monitor RPA process instances. This interface is generally graphical, but most systems also provide an API. Robots are programs that run on a physical or virtual machine. They execute RPA process instances, meaning that they emulate user interaction on the machine they run on. The controller distributes the jobs of running certain RPA process instances among a pool of robots that are connected to it.

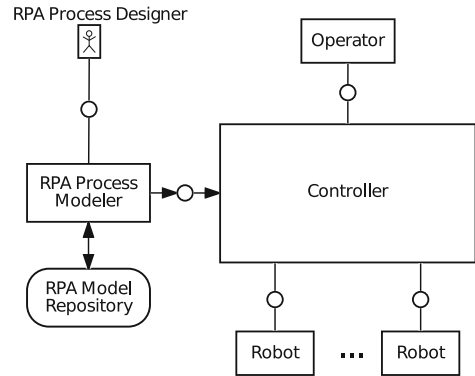


Fig. 3. RPAS architecture

Generally speaking, RPA is usually applied to processes that are too infrequent for traditional process automation to be profitable, but are still repetitious enough to be formalized into an RPA process model [1]. Several criteria for RPA applicability are widely accepted [1,2,5,6,8]: Process in- and outputs must be in a machine readable format. As machine recognition capabilities greatly increase due to upcoming advancements in AI technologies, so does RPA applicability. A process to be automated must be well-defined and have a low change rate. Otherwise, inconsistencies between process model and actual process inhibit RPA success drastically. In addition, a low decision complexity is required for RPA processes, as robots cannot (yet) fully replace human decision making.

In comparison to traditional process automation, RPA is cheaper to establish and provides a much faster return on investment [2,6,7]. RPA enables the automation of processes that could not be traditionally automated, as it does not require any APIs [5]. In comparison to human execution of processes, RPA saves lot of time. Together with the fact that RPA workforce can easily be scaled up, this allows for a much greater number of cases to be handled [5]. Additionally, robots execute processes consistently and avoid human errors, therefore increasing effectiveness and process compliance [6].

3 Problem Statement

Despite all benefits, RPA has strong limitations: In order to identify and implement an RPA process, extensive process knowledge is required. Existing work has shown that, if no such knowledge is available (e.g. no other systems for gathering it are in place), the benefits of RPA are far less significant, as much time

and effort has to be put into gaining that knowledge [5,6]. RPA is often considered a risk, due to the fact that it is hard to test and, once deployed, the robots execute a potentially faulty process at very high rate with high consistency [5,8]. Testing RPA requires setting up test environments for all software dependencies. While initial work has been done recently [4], no standard testing mechanisms have been established yet. In addition, whereas humans check each of their steps in each process instance, RPA has only few built-in error recognition algorithms [5,8]. The largest identified flaw of RPA is that its scope is too limited and therefore insufficient to manage and automate business processes on an organizational level. RPA orchestration is usually limited to orchestrating the robots, but does not allow to orchestrate large-scale or end-to-end processes [8] in the scope of a whole organization. RPA systems lack the ability to execute activities that are not automated with robots: Human elements cannot be completely removed from all processes of an organization, but RPA does not provide concepts to execute tasks that have to be done manually. Likewise, executing larger software and coordinating services is outside the scope of operating a user interface. Moreover, resource management is crucial for larger companies to maximize the efficiency of the employees and use their full potential. Contrary to BPM, RPA does not include research on this topic.

To cope with all the limitations of RPA, related work suggests to embed RPA into BPM [2,5,6,8]. However, the nature of this integration has not yet been investigated. A concrete integration approach needs to be developed to serve as basis for RPA projects and further research.

4 Related Work

Although not much work has been done on the integration of RPA into BPM, existing work on RPA often suggests that such an integration is desirable, and describes the relation between the two approaches.

One possible approach to enact RPA was described by Kirchmer et al. in [5]. They introduce the value-driven robotic process automation approach that includes criteria for the identification of processes which can be automated, and basic methods for design and deployment of RPA processes. Although they suggest to integrate this approach into the larger BPM context, they do not describe how to realize the integration or which benefits and synergies arise from this combination. They identify the need for BPM to be set up or already running in order to increase reliability and enable exception handling. Additionally, the need for the combination is motivated with process governance, which is provided by BPM and required for RPA. As entry point for further research and development, they suggest the integration of RPA into a larger automation context.

According to [8], BPM is a prerequisite for implementing RPA successfully. This is based mainly on capabilities that BPM has over RPA: BPM includes

monitoring and process improvement techniques and makes in-depth process knowledge available. This knowledge is required for RPA process identification and implementation. It can also be used for communication about the processes, allowing to attune activities on an organizational level.

Further work on the relation between RPA and BPM include the following: Kroll et al. state that RPA draws advantages from process standardization provided by BPM [6]. Aguirre et al. encourage the combination of RPA tools and BPM systems, demanding for further studies on it [2]. Lacity et al. describe a case study on the integration of RPA in a company, which states that establishing RPA profited a lot from having a BPMS running in the background [7].

While existing work presents methods for RPA process configuration and enactment, and motivates their integration into the BPM methodology, it does not describe how to accomplish the integration. Therefore, we propose below a solution to realize RPA processes inside a BPM context.

5 Architectural Integration

In this section, we propose an architecture to link RPA and BPM systems. This architecture aims to provide a technological foundation for RPA-to-BPM integration, give an execution context for RPA systems, and facilitate the implementation of RPA processes in an organizational setting.

It already is technologically possible to call an RPAS from a BPMS. For instance, REST calls can be sent from script tasks to RPAS APIs. However, these solutions expose implementation details of the link between the two systems. In order to avoid a massive development effort each time a company wants to automate an RPA process, implementation details should be encapsulated in BPMS abstractions. Therefore, this section aims at providing an architecture that allows the systematic use of RPA and BPM systems in tandem.

The integrated architecture specifies a system that acts as a bridge between the RPA and BPM systems. The system enables the instantiation and execution of a robotic-automated activity (by an RPAS) during the run of a higher-scope business process (by a BPMS) without the intervention of a human. Configured process models should not include the configuration of the actual implementation but only the configuration of the process to automate. Therefore, process designers only need to specify activity inputs and outputs and do not have to deal with technical details.

Assuming that both, RPA and BPM, systems are already set up independently and could run on their own, the system design conforms to the following rationales: It is non-invasive in that it does not limit the capabilities of either of the standalone systems. The design is also independent of specific BPMS or RPAS vendors. In order to minimize the effort for an organization, deploying the bridge should only include a single setup and configuration step. Regarding the separation of concerns between the RPAS and BPMS, each system manages what it is designed for. The system's design does not put constraints on the abstraction level of the processes that are to be automated. Rather, the organization decides on which business process abstraction level the RPA process shall

be implemented. We assume, however, that the top process level is managed by the BPMS.

The concrete design solution is depicted in Fig. 4. It introduces an additional component between the BPM and RPA systems: This bridge system connects to the BPMS as external application that the execution of a task can be delegated to. On the other side, it uses the API provided by the RPAS as interface to act as RPA operator. The bridge system is split into specific adapters for the BPM and RPA systems and a core system in between.

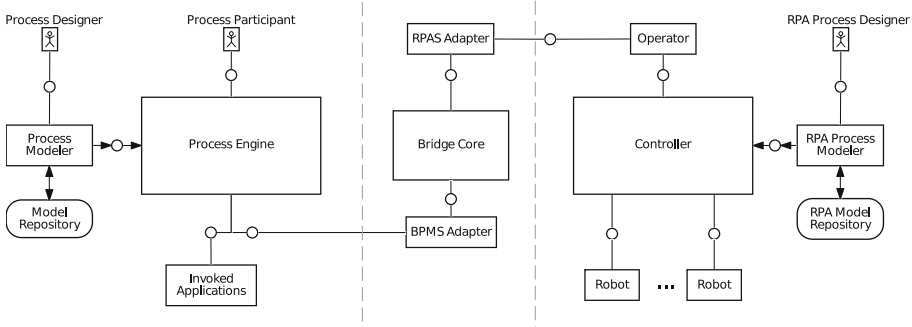


Fig. 4. Architecture of bridge system

In accordance with the rationales, we made the following design decisions: To guarantee the rationale of vendor independence, the bridge is composed of two interchangeable adapters, and a core system, which contains the functionality that can be used for all pairs of vendors. By using existing interfaces of the BPMS, namely the activity execution delegates, and of the RPAS, namely the operator API, the systems remain untouched. Thus, the bridge does not induce any constraints to them. The RPA controller is designed to orchestrate the robots and distribute jobs amongst them. Consequently, we preserve this responsibility, following the separation of concerns rationale. As a result, the single robots are unknown to the BPMS because the controller acts as a mediator. Furthermore, a BPMS offers the ability to define reactions to business exceptions and errors. Therefore, exceptions that occur during the execution of RPA processes are forwarded to the BPMS to be handled.

The system behavior for executing an RPA activity is depicted in Fig. 5 as an UML sequence diagram. The bridge is deployed into the BPMS as delegate for the execution of a defined type of activity. Therefore, the configuration of the activity to automate must include the information required to configure and execute the RPA process. Once the BPMS starts an activity instance of the specified type, it delegates the execution to a BPMS-specific adapter. This adapter converts the BPMS process information and activity inputs to a standardized RPA process input format. The core system distributes this information to an RPAS-specific adapter and governs the execution of the RPA process. This governing is

phased in starting the RPA-process, waiting for its termination, and retrieving the output. The RPAS adapter implements this governing interface, interacts with the robot controller via the controller API, and converts the RPAS-specific process results to a standardized RPA process result format. The retrieved results are first passed to the core and then to the BPMS adapter, which updates the BPMS process and its variables accordingly before terminating the activity.

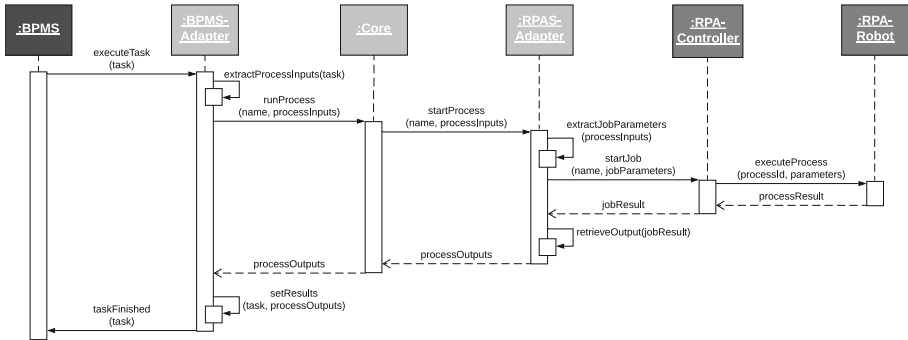


Fig. 5. Sequence of the execution of one RPA activity

6 Methodological Integration

This section addresses the research gap on how to embed RPA process configuration and enactment into the BPM methodology by combining the RPA methods with the BPM lifecycle. It further describes the synergies arising from this combination.

The resulting methodology provides a standardized approach for RPA-to-BPM integration. It allows to use existing techniques of BPM in order to design, configure, enact, and evaluate RPA processes. We approach the design of our methodology by transferring the existing BPM methodology to realizing RPA. The resultant RPA realization methods are then reintegrated into the BPM methodology.

The business process lifecycle described in Sect. 2.1 provides a detailed standard methodology for managing business processes on different levels of abstraction. We therefore chose it to be our base and adapt it to form an RPA-aware BPM lifecycle. As RPA systems can only automate processes on a low level of abstraction, RPA processes can be considered activities of a parent business process. They can therefore be handled as such and whenever operations on activities would be performed by the surrounding BPM framework, they are also performed on the RPA processes. This way, information and tools available to methods on the outer process are also available for the methods on RPA processes.

The adaptations to the BPM lifecycle (depicted in Fig. 6), which form the RPA-aware lifecycle, are structured as follows:

In the *design and analysis* phase, RPA processes are identified and modeled, following the criteria from Sect. 2.2. Important aspects for the identification are process repetitiveness, in- and outputs, and the necessity for human involvement. Information on these aspects is provided by techniques from the BPM design and analysis phase, allowing better judgment about the applicability of RPA to certain activities. Additionally, thorough process knowledge gained in past iterations is provided by the BPM evaluation phase, which eases the modeling of RPA processes, and gives leverage points for improving existing RPA processes. This design results in a semi-formal representation of the processes to automate, which can be validated against execution data and interviews gathered by BPM techniques.

In the *configuration* phase, an organization selects their RPA system to use, and how to deploy the robots. The robots can either run on physical or virtual machines. The semi-formal process representation is then implemented as an executable RPA process model. In the next step, the controller is set up and configured, the model is deployed to the controller, and the robots are installed. The existing infrastructure of BPM deployment can be utilized therefor. The set up RPA system must now be tested before being released. The configuration results in implemented robot process models and a running RPAS infrastructure.

In the *enactment* phase, the RPA processes are operated. The operators start RPA process instances via the RPA controller. In our embedded context, the operator role is adopted by the BPMS through the bridge system described in Sect. 5. As RPA process execution is prone to errors, it needs detailed monitoring during operation. The monitoring and exception handling mechanisms of the BPMS can be used to support this. In addition to the data collected by the RPAS, the BPMS gathers further information about the execution, for example, the duration of execution, or compliance and conformance properties. This helps monitoring by revealing when an RPA process does not behave as expected. Maintenance needs to be performed when bugs are detected or changing requirements exact immediate fixes. The enactment results in raw execution data.

In the *evaluation* phase, information about how the robot performed is derived from the raw data of the enactment, in order to use it as basis for the next (re-)design phase. For this, BPM provides access to its tools such as process mining, which has already been shown in related work [3].

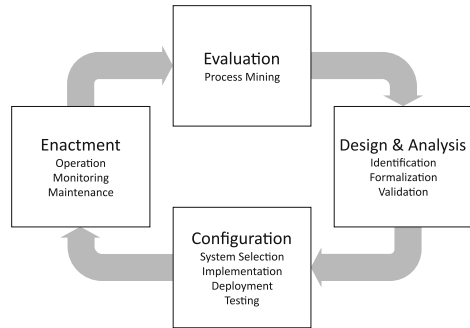


Fig. 6. RPA-supporting or -enacting methods of the RPA-aware BPM lifecycle

Summing up, the RPA-BPM technological integration imposes the need for a careful methodological integration, which is introduced in this section on the bases of the more mature area of BPM.

7 Evaluation

To evaluate the proposed overall BPM-RPA integration, we developed a prototypical implementation to evaluate the architectural integration and applied the methodological integration to an use case scenario. The evaluation section is concluded with discussion on the results.

7.1 Architectural Integration Feasibility

To prove the concept of the architectural integration we created a prototypical implementation as Java Maven library. The system is called Talos and is available on GitHub². Talos provides two interfaces, one for BPMS adapters and one for RPAS adapters. The Talos repository also includes an example implementation of those two interfaces using Camunda BPMN Workflow Engine³ as BPMS and UiPath Community Cloud⁴ as RPAS.

The prototype provides the *RPADelegate* class as implementation of a delegate for service tasks. When a service task that has been configured with this class is executed in Camunda, UiPath is automatically called to execute a specified RPA process. Thus, the only configuration required inside the process model is using this delegate and the RPA process identifier. This fulfills our main design goal of automatic delegation, while also not revealing bridge implementation details, hence retaining abstraction.

Alternative implementations of the interfaces allow a seamless change of BPM or RPA systems. Therefore, Talos is independent of vendors. To deploy the bridge, it is only necessary to download the latest build from the repository and deploy it to Camunda. For other BPM systems, there exist similar deployment mechanisms for applications to delegate the execution of tasks. Summarizing, the deployment effort can be considered as low.

To conclude, we showed how RPA processes can be technologically integrated into a BPMS with low effort and without violating process management abstractions.

7.2 Application to Use Case

This section provides an example application of our proposed integration solution in a company.

Assume a company which provides financial services on application. In the past, incoming applications were inserted into a web interface by a clerk. To

² <https://github.com/LeonBein/Talos>.

³ <https://camunda.com/de/products/bpmn-engine/>.

⁴ <https://www.uipath.com/de/>.

improve this time consuming, repetitive, and error-prone task, the company decides to introduce process automation with business process management.

The original process (depicted in Fig. 7) starts when an application form is received by mail. A clerk then manually validates the information contained in the application. If it is valid, he inserts all information into an online form. If the information is not valid, the clerk prepares a request for a valid application that is sent to the applicant afterwards.

Due to the introduction of BPM, the company now follows the BPM lifecycle:

During *design and analysis* phase of the original business process, it has been discovered. BPM process analysis via process mining and employee interviews has shown that the process is time-consuming, thus posing a bottleneck. Furthermore, employees tend to make typing mistakes, as the task is highly repetitious. Therefore, the process has proven to be error-prone.

To improve performance, the company has decided to automate the process as far as possible. As the original process has already been discovered and modeled, information like involved data objects is known. Text from scanned paper-based documents can be extracted with RPA [9] and entering data into web forms is one of the classical use cases of RPA. Therefore, the company decides to use RPA to automate the interaction with the incoming document and the web form used to submit the application to their system. Additionally, validating the data and preparing requests for valid applications are automated by means of traditional process automation.

The new process (depicted in Fig. 8) is structured as follows: As before, it starts when an application has been received by mail. The clerk scans the application to make it machine readable. One robot then uses text recognition to automatically extract the form data and writes it to a CSV file. If the robot cannot parse the text, the clerk has to create it manually. The CSV file is now validated by a validation service. If the data is valid, a second robot inserts it into the web interface automatically. Otherwise, a service is used to prepare a request for valid application that is then sent to the applicant.

In addition to the business process model, the company creates textual documentations as semi-formal representation of the robotic-automated tasks.

In the *configuration* phase, the company selects an RPA provider and sets the system up. Robots can run in virtual machines on the same server infrastructure where the BPMS is put in place. Talos is deployed to the BPMS as execution delegate and configured to use the specific RPAS for task execution. The company then implements the redesigned process in their BPMS, configuring Talos

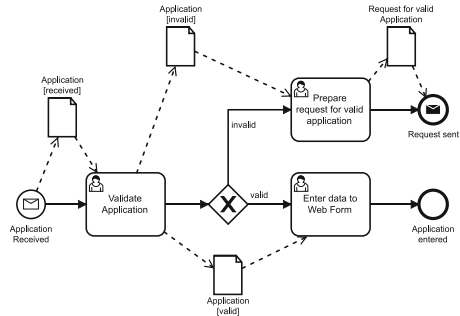


Fig. 7. Original application process without RPA

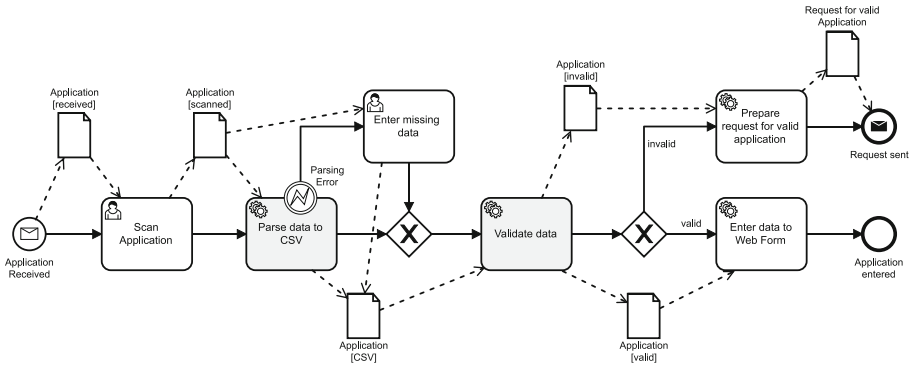


Fig. 8. Application process with RPA. Tasks that are automated with RPA are coloured gray

to handle the execution of the designated RPA tasks. The RPA task models that are used are implemented with the corresponding RPA process modeler. The business process model can then be deployed to the BPMS the same way as if RPA was not introduced.

Together with Talos, the BPMS assumes the operation of the RPAS during the *enactment* phase. If the robot cannot parse the document, the system allows to switch back to manual execution, thus putting the exception into a process context. Process monitoring additionally helps the company to ensure that the robots behave as expected.

During *evaluation* phase, logs of process executions are analyzed with process mining. This includes data on the execution of the RPA tasks, allowing to measure the time improvements compared to the original process and also to detect if further optimization is needed. The company also identifies more processes which should be improved and are potential candidates for RPA solutions.

7.3 Discussion

The main benefit of the RPA-aware BPM lifecycle is that it describes a concrete approach for realizing RPA in a BPM context, addressing the research gap. This allows RPA to be introduced to and used in organizations more easily, especially when BPM infrastructure and knowledge are already in place. In contrary to ad-hoc usage, not every organization needs to define their own methodology. Instead, they can build on the RPA-aware BPM lifecycle. Together with further research, the organizational experience can help to improve the methodology. Using a proven set of methods decreases the risk of project failure for the organizations [5]. As BPM techniques and technologies are well tested and therefore efficient and stable, these risks are further reduced. Furthermore, BPM scales well, allowing to handle numerous processes. The RPA-aware BPM lifecycle also provides the means to handle multiple RPA processes, as they are fully integrated. The fact that RPA scales with BPM renders RPA scalable as well.

The RPA-aware lifecycle deals with the shortcomings of RPA:

The knowledge created during the BPM lifecycle is necessary or at least beneficial for RPA. This minimizes the overhead created when running both systems concurrently, as the process knowledge needs to be gathered either way. BPM provides information about process attributes, inputs, and outputs in the analysis phase and about the execution in the enactment phase. As shown in the example, this information can be used to identify and model new RPA processes and to improve existing ones. In addition to process optimization, the BPMS provides documentation and standardization for all processes not managed by the RPAS.

The RPA-aware lifecycle also helps to cope with the RPA faultiness. BPM exception handling provides a standardized and well-tried approach to handle business exceptions, thus unburdening employees. BPMS monitoring accelerates finding errors in the RPA processes. Technical checking mechanisms incorporated into the BPMS catch execution errors and provide basic handling for them. Data gathered from enactment phases can be used to improve RPA processes in further iterations, thus making the processes more stable and eliminating errors [8].

Integrating the RPAS into a BPMS complements the capabilities an RPAS lacks. High level process orchestration, the execution of non-RPA activities, resource management, process monitoring, process mining, and additional features are assumed by the BPMS and its periphery tools.

Despite the benefits, some aspects of RPA-to-BPM integration are not yet covered. While this solution provides means for an integration on the software architectural and methodological level, it does not consider change management like employee training and changes on an organizational level. The example company must convince the clerks of the change and find new occupations for them. The given approach also does not build up a solution tailored to the needs of RPA. Therefore, some RPA specific issues might not be examined and addressed. An example is the issue of RPA process testing, which requires the development of specialized environments. Additionally, the lifecycle does not provide new RPA-specific error and exception handling mechanisms. Furthermore, this solution is limited in the level of detail it provides for each phase. For instance, no pre-implementation formalization approach is given for the design of RPA processes.

8 Conclusion

An RPA integration into BPM can break up RPA limitations and provide the process knowledge required for a successful RPA realization. In this paper, we presented an integration solution, which provides the basic means to implement RPA processes inside BPM environments. Our solution consists of a software architectural bridge and an RPA-aware BPM lifecycle which link the RPA and BPM systems and integrate their methodologies. The architectural bridge has been evaluated with a working prototype that allows the technological integration of RPA processes into a BPMS with low effort, preserving business process

management abstraction. The RPA-aware BPM lifecycle describes RPA methods that are embedded to the larger BPM discipline, thus profiting from synergies drawn from BPM tooling and methodology.

With our work, we integrate RPA into the larger BPM automation scope. Thereby, we follow the research proposal of Kirchmer et al. [5]. In return, their RPA process configuration and enactment methods can be used to provide another view on the respective RPA lifecycle phases. We discussed a further approach to combine RPA and BPM, providing foundations for further studies, as proposed by Aguirre and Rodriguez [2].

Future work on the exemplary bridge implementation includes realizing concepts to catch RPA process execution exceptions and forward them to the BPMS to handle them. Future research should explore the impact of integrated RPA on an organizational level, including change management and workforce training. For example, according to [5], inappropriate preparation of employees may lead to a significant reduction of the efficiency benefits gained by introducing RPA. It may also be worth to investigate approaches for standalone RPA that are specifically tailored to it, and to compare these to our solution. Dedicated testing strategies need to be examined and developed to further decrease the error rate of RPA process executions. For the RPA-aware lifecycle, much work can be done by exploring the details of its phases, further improving it, and defining it more precisely.

References

1. van der Aalst, W.M.P., Bichler, M., Heinzl, A.: Robotic process automation. *Bus. Inf. Syst. Eng.* **60**(4), 269–272 (2018). <https://doi.org/10.1007/s12599-018-0542-4>
2. Aguirre, S., Rodriguez, A.: Automation of a business process using Robotic Process Automation (RPA): a case study. In: Figueroa-García, J.C., López-Santana, E.R., Villa-Ramírez, J.L., Ferro-Escobar, R. (eds.) WEA 2017. CCIS, vol. 742, pp. 65–71. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66963-2_7
3. Jimenez-Ramirez, A., Reijers, H.A., Barba, I., Del Valle, C.: A method to improve the early stages of the robotic process automation lifecycle. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 446–461. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_28
4. Jiménez-Ramírez, A., Chacón-Montero, J., Wojdyski, T., González Enríquez, J.: Automated testing in robotic process automation projects. *J. Softw. Evol. Process* e2259 (2020). <https://doi.org/10.1002/smr.2259>
5. Kirchmer, M., Franz, P.: Value-driven Robotic Process Automation (RPA). In: Shishkov, B. (ed.) BMSD 2019. LNBIP, vol. 356, pp. 31–46. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24854-3_3
6. Kroll, C., Bujak, A., Darius, V., Enders, W., Esser, M.: Robotic process automation - robots conquer business processes in back offices (2016). <https://www.capgemini.com/consulting-de/wp-content/uploads/sites/32/2017/08/robotic-process-automation-study.pdf>
7. Lacity, M.C., Willcocks, L.P.: Robotic process automation at telefónica O2. *MIS Q. Exec.* **15**(1) (2016). <http://misqe.org/ojs2/index.php/misqe/article/view/620>

8. Signavio Whitepaper: Putting the ‘P’ in RPA: overcoming the challenges of RPA Implementation. Technical report, Signavio GmbH (2019). <https://www.signavio.com/downloads/white-papers/overcoming-challenges-rpa-implementation/>
9. Tornbohm, C.: Market guide for robotic process automation software (2017). [Gartner.com](https://www.gartner.com)
10. Weske, M.: Business Process Management, 3rd edn. Springer, Heidelberg (2019). <https://doi.org/10.1007/978-3-662-59432-2>