



Migrating from RPA to Backend Automation: An Exploratory Study

Andre Strothmann^(✉) and Matthias Schulte

viadee Unternehmensberatung AG, Münster, Germany
{Andre.Strothmann,Matthias.Schulte}@viadee.de

Abstract. Robotic Process Automation (RPA) is often considered a short-term solution that bridges technology gaps temporarily. However, it is no longer the choice of automation technology when backend automation becomes readily available. We collect insights from thirteen interviews and two literature reviews to investigate the migration from frontend automation with RPA towards backend automation with APIs. This paper explores how the robots need to be designed and prioritized so that they can be easily replaceable in the order that allows an organization to benefit the most from a change in the automation infrastructure. Findings encompass requirements for the design of migration-ready RPA bots, challenges to overcome, and selection criteria for prioritization. We integrate these results into Business Process Management (BPM) from a methodological as well as architectural perspective. The practicability of our findings is backed by a prototypical implementation with which we showcase the migration from frontend to backend automation in an organizational setting from the telecommunications industry.

Keywords: BPM · RPA · APIs · Backend Migration · Design Science

1 Introduction

A vast majority of consumers interacts almost daily with organizations whose legacy infrastructure does not provide open and accessible interfaces. Therefore, many companies move towards Robotic Process Automation (RPA) to tackle the automation of processes using application systems that cannot be integrated otherwise. While RPA is considered useful for rapid automation of these hard-to-integrate components, it is no longer the automation technology of choice once the historically grown systems are given the opportunity of application-internal process automation [2, 16]. This requires gradually retiring the robots to replace them with services accessible via Application Programming Interfaces (APIs).

Although RPA has gained momentum in practice, literature does not address how the migration from frontend towards backend automation can look like. Failing to modernize the infrastructure with a more sustainable approach to automation, however, may lead to technical debts for the additional rework at a subsequent time. In response to this research gap, we conduct an exploratory

study to analyze how RPA bots can be transitioned towards APIs. First, the design of migration-ready robots is explored so that their latter replacement can happen with ease. Since selecting the wrong candidates can cause automation projects to not deliver the expected results [17], the second goal is to provide strategic foresight in prioritizing robots that allow an organization to benefit the most from a change in the automation infrastructure. Lastly, we implement a prototype to demonstrate the methodological and architectural results regarding the migration-readiness of robots and their prioritized replacement in an industry setting. Thus, the analysis is guided by two research questions (RQs):

1. *How can the development and design of RPA robots be approached when their migration towards APIs is the long-term goal along the automation journey?*
2. *In which order should the migration-ready RPA robots be prioritized so that an organization can get the most out of their transition towards APIs?*

2 Preliminaries

2.1 Business Process Management

BPM provides concepts, methods, and techniques to iteratively improve organizational performance in the form of a lifecycle with circular dependencies [18]. This continuous cycle is structured into the six phases from Fig. 1a [4].

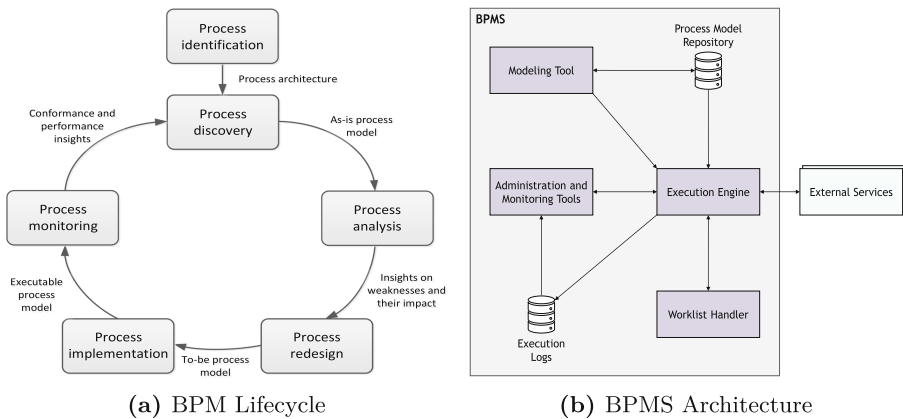


Fig. 1. BPM Lifecycle and BPMS Architecture [4]

The entry point is the phase of (1) *process identification*. A relevant set of processes is structured into an overall architecture to identify important relationships. Subsequently, (2) *process discovery* intends to document these processes and map the current state on how activities are performed. This typically yields as-is models that are used as input for (3) *process analysis*. The outcome is a collection of existing problems, whose impact is quantified to prioritize which issues

should be addressed first. Based on this prioritization, processes are transformed into to-be models during (4) *process redesign*. Due to an abstract design, the to-be models are systematically reworked into executable workflow models along the (5) *process implementation*. Once the redesigned processes are operated, the phase of (6) *process monitoring* is aimed at collecting execution data, observing deviations, and providing the gained knowledge for the next iteration.

Enterprises that practice BPM along this lifecycle may achieve additional benefits if process-aware information systems are used [18]. Such systems are called Business Process Management Systems (BPMSs) and follow the generic architecture from Fig. 1b [4]. This architecture encompasses a *modeling tool* for visualizing process descriptions in terms of their constituent activities. The resultant process models are stored in a *process model repository* so that the *process engine* can pull them for execution. The process engine is responsible for executing the process and distributing the units of work among resources. In case these work items await user interaction, the process engine places them on to worklists for attention by the *worklist handler*. In case the work items require to be completed automatically, *external services* can be consumed that operate outside the realm of the BPMS. This and other execution data is what the process engine keeps *execution logs* of. It allows *administration and monitoring tools* to provide statistics for evaluation with respect to the intended behavior.

2.2 Robotic Process Automation

RPA is a lightweight technology to automate the point of interaction between users and legacy systems [7]. This is realized by mimicking the streams of clicks a user performs in the frontend to provide rapid automation of monotonous activities. It can be used for a broad spectrum of processes, covering a range from single tasks to the automation of entire business processes [1,4]. However, the identification of suitable candidates for automation is crucial and an important step in the beginning of an RPA project [9]. Thus, criteria need to be established to determine which activities are worth automating using RPA [15]. Promising candidates become documented in terms of their behavior in the user interface, developed into RPA bots, and deployed to the respective environment. Subsequently, the robots are tested to ensure they work as expected and monitored to assess the performance in production. As shown in Fig. 2a, these outcomes can be used to enter a new identification cycle for continuous improvement.

In an organizational setting, RPA requires certain architectural necessities to be successfully enacted. As described in Fig. 2b, RPA bots need to be orchestrated with the help of a so-called *control system* to manage their execution [11]. This controller holds the repository of the process models that RPA process designers create through interaction with a modeling tool [11]. However, it also distributes the jobs of the process instances among a collection of RPA bots, and exposes both graphical surfaces and programmatic interfaces that an operator can use to start and supervise the ongoing and completed automation jobs [11].

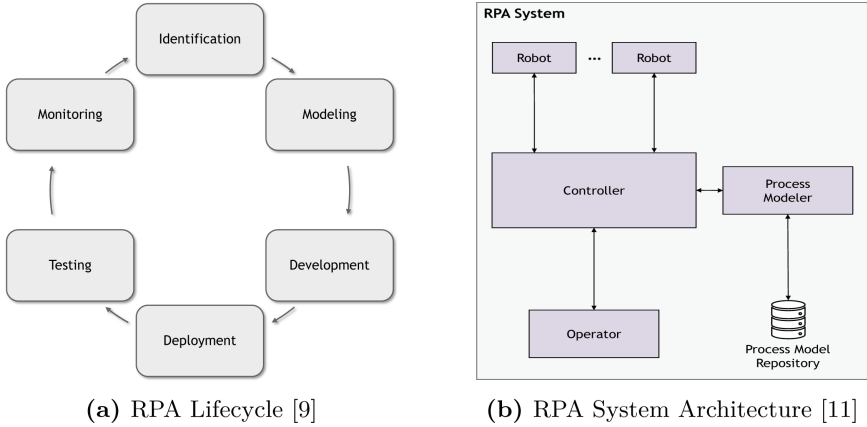


Fig. 2. RPA Lifecycle and RPA System Architecture

3 Methodology

We intend to close the identified research gap with an artefact that is beyond current domain knowledge. To address this objective systematically, the design science research method proposed by HEVNER has established itself as an appropriate approach [8]. For this reason, we structure our research along three related cycles of activities: relevance, rigor, and design.

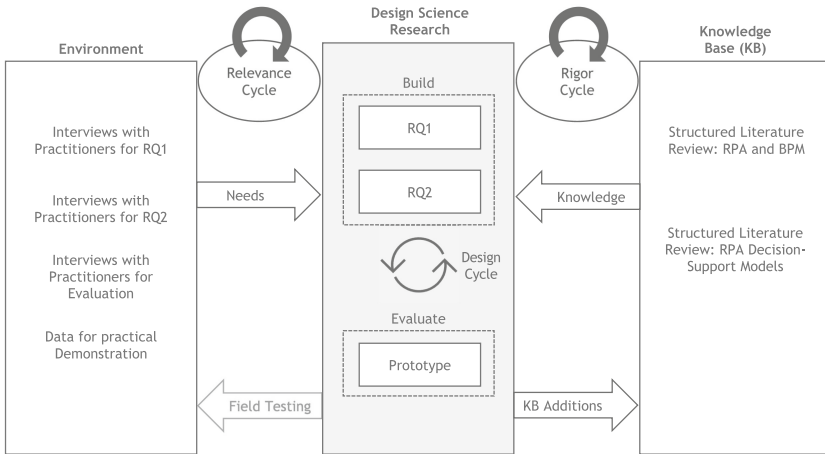


Fig. 3. Adopted Approach to Design Science [8]

Figure 3 presents these cycles together with their circulated domains. As for the first step, we gather input on the basis of interviews with subject-matter

experts in an exploratory manner. The collected insights are used to inform the structured literature reviews so that our findings from practice are backed with theory. Subsequently, we conduct the build phase to suggest an initial design for the first and second research question and evaluate its degree of suitability through another round of qualitative interviews. Continuing the evaluation, the improved design is put into a prototype to showcase the migration from frontend towards backend automation in a sample scenario from the telecommunications industry. An evaluation in field conditions remains out of scope for this study.

Table 1. Overview of Interview Partners

Stage	Sector	Organization	Position
1	Insurance	Gothaer Versicherungsbank	Head of RPA
	Financial Services	Commerzbank	Head of Automation
	Engineering	thyssenkrupp Material Services	Project Manager
	Education	Fachhochschule Südwestfalen	RPA Researcher
	Telecommunication	Deutsche Telekom	Vice President IT
			Project Manager
		NetCologne	System Engineer
	IT Consulting	IBM	Platform Architect
			Solution Specialist
Camunda		Chief Technologist	
Almato		Director Automation	
2	WDW Consulting Köln	CEO	
	viadee Unternehmensberatung	Managing Consultant	

For the exploratory analysis, we approached 11 experts from various industries. A semi-structured technique is selected as the format of the interviews to deviate from the predetermined script whenever we feel the responses require further examination [13]. We stopped the inclusion of additional cases the moment data saturation was reached. The second interview round is conducted with another two process management experts to reassure the rigour of the research. We stick with an unstructured approach to ensure the respondents are not biased by questions set beforehand, rather can pursue the improvement opportunities they feel are important to contribute to an enhanced design [13]. Table 1 provides an overview of sectors, organizations, and positions of all thirteen interviewees we talked to. The choice of respondents is done on the basis of our own judgment and primarily based on hands-on experience with RPA as the leading selection criterion. The interviews took an average of 45 min, were conducted online, and have been analyzed in deductive-inductive manner using the approach by KUCKARTZ for qualitative text analysis [10].

The findings from the first interview round motivate to look up literature at the intersection of RPA and BPM in terms of RQ1. Regarding RQ2, insights reveal the need for consulting decision-support models for the identification of

candidates that are suitable for an automation with RPA. For both structured literature reviews, we guide our analysis along the framework proposed by VOM BROCKE ET AL. [3] and query data from five leading scientific databases, namely *Scopus*, *IEEE Explore*, *SpringerLink*, *Web of Science*, and *ACM Digital Library*.

From 216 (166) studies that fit the search string for RQ1¹ (RQ2²) initially, 185 (133) are excluded after reading their abstracts. This results in 31 (33) publications that remain for full-text reading, out of which 24 (19) are kept. Without duplicates, this leads to a total number of 19 (11) publications. Through backward referencing, additional 5 (3) articles are identified to be relevant. This gives an overall amount of 24 (14) studies that are left in the final review set.

4 Exploratory Analysis

4.1 Migration-Ready RPA Bots

The analysis of interview data uncovers requirements to the design and challenges to expect during migration. These are summarized in Table 2 and presented next.

Table 2. Design Requirements (DRs) and Migration Challenges (MCs)

ID	Description
DR1	Bots need to automate single tasks on API-granularity in the process
DR2	Bots need to be orchestrated to separate bot- from process-logic
DR3	Bots need to provide automation within the boundary of single systems
DR4	Bots need to be implemented for reuse across processes
DR5	Bots need to be idempotent to realize the same error behavior in the process
DR6	Not only the bot, but also the process logic needs to be designed such that it allows the task to be automated independent of the use of RPA or APIs
MC1	The input and output data between bots and APIs may deviate

Interviewees claim that robots automating tasks within the process are easier to be replaced compared to robots automating the business process itself (cf. DR1). These tasks need to bind modular functionality so that calling a robot to run an operation is conceptually the same as calling an API to realize service behaviour. This requires keeping the process layer apart from the bot logic: instead of confusing a robot’s clicks with the essential parts of the process, its activities in a system’s user interface should be encapsulated into a service abstraction that is organized along a consolidated workflow (cf. DR2).

It is also recommended to not let a robot interact with multiple systems at once (cf. DR3). Since APIs are built for exposing capabilities from a single core systems, RPA bots should not exceed system boundaries as well. Similarly,

¹ (“Business Process Management” OR BPM) AND (“Robotic Process Automation” OR RPA).

² (“Robotic Process Automation”) AND (“Selection” OR “Identification” OR “Decision Support”).

interviewees consider the behaviour on error to be another important factor that needs to be aligned among both approaches to automation (cf. DR5). An API is usually idempotent so that the executions can be repeated without unintended side effects. If RPA bots are integrated into the business process following a different design paradigm, transitioning may become harder as non-idempotent robots may introduce additional process logic that would need to be removed before an API can take over the automation.

Interview data also reveals that the likelihood for a robot to be replaced increases with its reusability across processes (cf. DR4). In case a robot is used only in a single workflow, the migration becomes difficult to justify from a cost perspective. An adaptive design, therefore, allows the transition to pay off faster.

However, the migration capability is said to be not only concerned with the design of the robots themselves, but also with the design of the orchestrating process. Accordingly, the process needs to be implemented so that the business functionality can be handled independently of the choice of the automation technology (cf. DR6). For instance, a process that is designed to always expect an ordered sequence of characters upon a task's completion is going to break in case the API provides the output as an integer value. This requires to refrain from incorporating hard-coded logic that applies for one of the approaches exclusively.

Along the same line of thought, the domain experts expect a major challenge the moment the transition process is triggered. This is due to the fact that the data between frontend and backend is not fully synchronized. Robots may require data as input or output fields that are distinct to the parameters passed as options with the endpoint and vice versa (cf. MC1). This requires the parameters between RPA bots and APIs to be mapped before the migration can begin.

4.2 Migration-Relevant Criteria for Prioritization

Interviewees suggest the prioritization along certain attributes (cf. Table 3). While we discuss their operationalization in Sect. 5.3, this section organizes them into *system characteristics*, *robotic performance*, and *task characteristics*.

In terms of the *system characteristics*, the user interface's stability and the expected end of life are mentioned as factors impacting the candidacy decision. A system with frequent changes to the frontend requires the robot to be reworked regularly. APIs, in contrast, are not sensitive to modifications on the presentation layer and should be the preferred pick for automating tasks that involve applications with many changes to the design. Similarly, a system going out of service in the near future may cause robots to become obsolete. To avoid an existing automation to be unexpectedly hit by the system's shutdown, it is suggested to preferably migrate robots that have to be replaced anyway soon.

In terms of the *robotic performance*, many interviewees posit that the amount of time it takes for a robot to fulfill a service provides yet another indication to derive the urgency for transition. If an RPA bot creates long customer waiting times, the company's outside perception is negatively affected and customers may switch to a competitor. The same holds true for both the rate and quality with which the robot performs the automation. Since APIs usually outper-

form RPA bots in terms of execution time, automation rate, and reliability, a migration should be initiated whenever low latency levels, a high degree of fully automated cases, or an adequate output quality cannot be guaranteed. Another criterion linked to robotic performance is the number of bot runners, i.e., dedicated machines needed to schedule automation jobs against runtime environments. Whenever the amount of bot runners is detected to be high, practitioners recognize their replacement to hold opportunities for potential cost savings.

The horizontal scale of those machines is determined by the number of times the task to be automated is executed. The higher the transaction amount the more bot runners are needed to perform well under an expanding workload. Thus, APIs are found to be the better fit in situations where the volume with which the task requires execution is extraordinary high.

Table 3. Criteria for the Prioritization of RPA Bots

In Favour of a Migration?	Yes	No
<i>System Characteristics</i>		
Frontend Stability	Changing	Stable
End of Life	Soon	Permanent
<i>Robotic Performance</i>		
Customer Waiting Time	High	Low
Automation Rate	Low	High
Quality of Results	Inadequate	Adequate
Number of Bot Runners	Many	Few
<i>Task Characteristics</i>		
Execution Frequency	Very High	Moderate to High
Number of Systems	One	Multiple
Business Impact	High	Low
Regulatory Compliance	Yes	No
Frequency of Reuse	High	Low

Further *task characteristics* are related to the business impact, the degree to which it can be reused, and the number of systems it interacts with. Particularly, business-critical tasks should be automated with more resilient solutions so that the risk of failure is minimized. Thus, an RPA bot automating a mission-critical activity is more urgent to migrate than a robot automating low-value work. In case this activity is also reusable across various verticals of the organisation, the robot should move up even further in the prioritization hierarchy since a multitude of cases could take advantage of the provided benefits. However, the API-based approach should not be chosen whenever the task encompasses swivel-chair operations that move data of one closed system into another. Since the benefits of RPA take effect all the more when the number of systems to access

increases, transitioning a robot that interacts with a single system provides more value compared to a replacement of a multi-system robot.

Finally, practitioners suggest to refrain from automating tasks with RPA in case regulatory constraints exist. If the company has to demonstrate compliance with legal standards relevant to the automated business function, additional security risks may arise due to the use of data that comes from the frontend.

5 Integration of Findings

5.1 Methodological Implications

In order for companies to undertake a gradual modernization of the automation infrastructure, the analysis has shown that interview partners recommend to approach the design of migration-ready RPA bots task-wise and within processes from start to finish. This would allow to replace these fine-granular robots with more robust API operations on the same granularity level, but would require an orchestration layer that handles RPA bots as part of the end-to-end process in the same way as service-oriented APIs. Literature studies this integration from a methodological perspective through a combination of the lifecycle approaches from both RPA and BPM [5, 11]. Particularly, FLECHSIG ET AL. extend the BPM lifecycle to consider RPA for processes that have not been worth automating through traditional BPMS [5]. While this approach adopts RPA for process automation in entirety, KÖNIG ET AL. introduce another model that considers RPA activities as individual tasks [11]. They assume the phases of the RPA lifecycle to be implicitly covered by the surrounding BPM framework, and allow the management of RPA processes on a different level of abstraction with the information available to the higher-scope workflow. This way, one layer exists that takes control of the parent process itself, while another layer micromanages RPA on task-oriented granularity along the automation chain.

Since this approach comes close to the requirements interviewees have posed, we choose this model as our base from which we start building. Thus, we follow the idea of using the existing lifecycle model to BPM, but introduce additional layers for RPA phases, DRs, and MCs to become explicitly linked to the phases of the overall framework from Fig. 1a. Furthermore, we need to revise the lifecycle management of RPA bots to make their potential retirement explicit. For this reason, the RPA lifecycle from Fig. 2a is extended so that robots can move into a migration phase after their performance is monitored.

These modifications lead to the four-layered RPA-aware and -migrating BPM framework that is visualized in Fig. 4. It adopts a subset of phases from the BPM lifecycle as the top-level layer and organizes RPA phases, DRs, and MCs underneath. Interviewees suggest to further distinguish DRs and MCs from a business and technical perspective so that technical matters can be explicitly demarcated from aspects that concern the migration's value proposition. Thus, DR4 is found to be not directly contributing to the migratability of a robot itself, rather motivating its replacement from a purely economical standpoint.

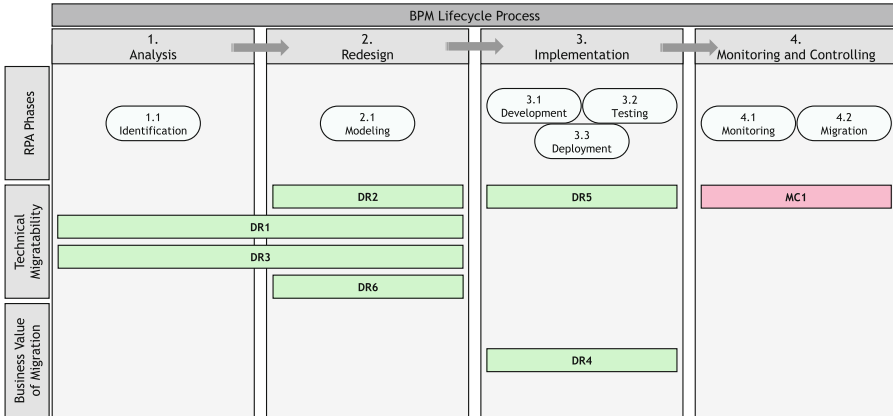


Fig. 4. Framework for the Development of Migration-Ready RPA Bots

The framework’s first step encompasses an analysis of an as-is process model through an identification of opportunities for robots to take over tedious tasks that are unable to be automated in the status quo. These tasks should be cut from granularity so that they encapsulate business logic that a service could realize within the boundaries of a single system (cf. DR1 and DR3). Secondly, the process is adapted to convert the manual activities into automated tasks that are part of an outer workflow (cf. DR2). Special care should be taken to ensure this workflow does not include hard-coded process logic that applies for RPA bots or APIs exclusively (cf. DR6). On the RPA layer, the redesign involves capturing the streams of clicks that are necessary to expose modular functionality from the level of the user interface. The resultant RPA model is developed into an idempotent and reusable robot (cf. DR4 and DR5), tested against errors, and deployed to an RPA system in a third step. The overarching workflow, on the other hand, is transformed into an executable process model that is deployed with the help of a BPMS. Since the execution of robots is generally decoupled from the actual orchestration, the BPMS is supposed to hand over control to the RPA system once the process moves on to an RPA-automated activity. The last step is about monitoring process performance. It includes the migration of robots as soon as the need for changing the automation infrastructure is recognized. This requires to prioritize candidates on the basis of their urgency for potential changeover (cf. RQ2), as well as to address the challenge of connecting distinct data models for a substitution with a minimum of effort (cf. MC1).

5.2 Architectural Implications

The development framework implies the orchestration of RPA bots at runtime of a top-level process. In order to ensure the migration-ready RPA bots can be executed as part of this outer workflow, literature argues that BPM and RPA systems need to be integrated so that the units of work can be delegated to a

robotic worker whenever needed [6, 11]. For this purpose, KÖNIG ET AL. provide an architectural blueprint that uses a BPMS for process orchestration, an RPA system for task automation, and a bridge system that joins them together [11]. Particularly, the bridge system serves as external application that the execution of RPA-automated activities can be delegated to. It forwards the request to the RPA system, waits for the robot to complete the assigned work in the frontend, and returns the result back to the BPMS. While we adopt the idea of having another component that provides access to functionality through the means of a mediator, RQ2 and MC1 yield further requirements that directly translate into architectural necessities. In terms of RQ2, this means that an architectural component is required with which the criteria from Table 3 can be evaluated to rank the robots in the order of their replacement. MC1, on the other hand, motivates the provision of a data mapping so that the transition can happen with the least possible changes to the process logic. These considerations come together to form the migration-enabling architecture depicted in Fig. 5. While we choose *Camunda BPM* as BPMS and *UiPath* as RPA automation software, the architectural idea remains equally feasible with other providers as well.

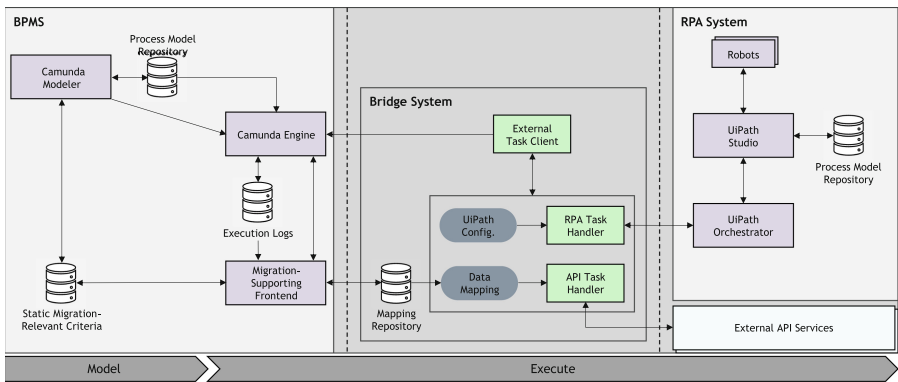


Fig. 5. RPA-Migrating Architecture

Due to the platform’s ability to accept extensions, we customize the *Camunda Modeler* so that the static migration-relevant criteria can be evaluated in the time of modeling. This allows the *Migration-Supporting Frontend* to query the criteria from a separate repository and to provide a suggestion for prioritization.

The *External Task Client* is used for realization of the bridge system. As a middleware, it polls the API of the *Camunda Engine* and fetches external tasks to provide them onto worklists. So-called *Task Handlers* pull these work items and process them in asynchronous manner. For the purpose of a migration, two different classes of task handlers are introduced. The RPA task handler is interested in pulling jobs that are meant to be automated by means of robots. It establishes a connection to the RPA system on the basis of access configurations

and returns the result upon completion. The API task handler, on the other hand, takes over the automation of external tasks that were formerly intended to be processed by use of RPA. To ensure it can receive and return data in the same way as the robot, the user has to provide mapping templates so that fields from one source can be connected to fields in another source. Thus, the migration from frontend to backend automation becomes a task of publishing the units of work no longer to the worklist the RPA task handler is committed to perform the automation for, rather to the queue the API task handler subscribes to.

5.3 Decision-Support Model for Candidate Selection

Reviewing literature reveals four steps that guide the selection problem between RPA bots and APIs: (1) finding criteria, (2) deriving weights, (3) determining an assessment scale, and (4) collecting data [12, 14]. While the migration-relevant information is already defined in Table 3, we extend the architecture from Fig. 5 so that weights can be assigned based on each criterion’s relative importance. Thus, we refrain from defining rigid values ourselves, but introduce an additional feature in the *migration-supporting frontend* with which the users can determine a weighting according to their particular needs. In terms of the baseline for assessment, we use a four-numbered likert scale for a clear direction of judgement upon which the degree of applicability of each criterion is measured. Thus, let $v_{jR_i} \in \{1, 2, 3, 4\}$ denote the rating value of criterion j for the RPA bot R_i , whereby j is an element of $\{1, 2, \dots, N\}$ and i a member of $\{1, 2, \dots, M\}$. With $N = 11$ representing the total number of migration-relevant criteria, $M \in \mathbb{N}$ referring to the overall quantity of robots, and $w_j \in \mathbb{Q}^+$ describing the weighting, Eq. 1 provides the prioritization value P_{R_i} for each RPA bot that exists. The higher this value is, the higher the urgency for a robot to be migrated.

$$\forall i \in \{1, 2, \dots, M\} P_{R_i} = \sum_{j=1}^N (w_j \cdot v_{jR_i}) \quad (1)$$

The collection of data happens either manually through interaction with the modeling tool or automatically by querying the workflow engine. This is, because the criteria can be dynamic or static in nature. Dynamic criteria do not need to be evaluated in advance, rather require evaluation during the execution of the workflow. A robot’s automation rate, for instance, should not be assessed when modeling, but needs to be evaluated with each instantiation of the process. This is different with the criteria whose degree of applicability remain static with each new occurrence of the process definition. Citing the examples of the number of systems or the existence of regulatory constraints, it is sufficient to do the evaluation once and neglect the examination at uniform intervals of time.

6 Case Study

We apply the framework for migration-ready RPA (cf. Fig. 4) together with a telecommunications company using the example of a line diagnosis process. As

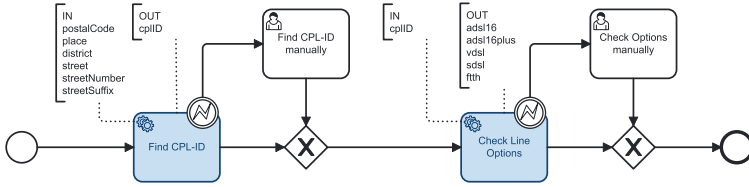


Fig. 6. Redesigned BPMN Diagram of the Line Diagnosis Process

for the first step, we conduct the *process analysis* through an identification of human tasks that are urgent to be robotic-automated. In terms of DR1 and DR3, these activities are decomposed to encapsulate service behavior along the axes of information systems. In this particular case, this includes the tasks of searching the Customer Precise Location-ID (CPL-ID) and assessing various options for the delivery of an internet connection in the designated region. During *process redesign*, both of these user activities are converted into service tasks that are part of an orchestrating workflow (cf. DR1). To further ease migration, we ensure the workflow does not bother with logic dependent on the choice of automation technology (cf. DR6). As a result, Fig. 6 depicts the process after redesign.

With the help of the *Camunda Modeler*, the RPA-configured activities are evaluated in terms of the migration-relevant criteria that is static in nature. For instance, the robot for the search of the CPL-ID is assumed to perform the automation within a system whose user interface is subject to regular change. This assessment can be found in Fig. 7a together with the specification of a topic that an *RPA Task Handler* can use to identify tasks of the same type.

In the next step, the *process implementation* takes place. For this purpose, we develop the RPA bots following DR4 and DR5. While either the search of the CPL-ID and the check of the line options do not manipulate a target resource in the database, the property of idempotency exists by default and does not require further attention during implementation. As for DR4, we make sure that both RPA bots can cover cases that are relevant apart from the process under consideration. For instance, the robot automating the check of the line options is given functionality that allows to not only return the availability of DSL options, but which can also evaluate the status of wireless technologies such as LTE.

Once the RPA bots are deployed to production in reusable and idempotent fashion, the *Camunda Engine* can trigger their execution as soon as the workflow arrives at an RPA-configured task. The *RPA Task Handler* subscribing to the queue called *FindCPLIDRobot*, for instance, reads data from the process context (postal code, place, district, street, number, and suffix), forwards it to the *UiPath Orchestrator*, and provides the result from external processing as CPL-ID.

During *Process Monitoring*, both static and dynamic criteria are evaluated. Assuming the robot that searches the CPL-ID is more urgent to migrate, MC1 is tackled in the belief an API exists that provides the CPL-ID with input fields differing by name. As shown in Fig. 7b, we establish a mapping template so that the data can be transformed as per the expectations of the technical

process design. By changing the topic of the service task from *FindCPLIDRobot* to *FindCPLIDApi*, the work items are no longer picked up by the RPA task handler, but can be claimed from the API worker in the exact same manner.³

(a) Evaluation of Criteria

Criterion		Assessment			
		1	2	3	4
System Characteristics					
Frontend Stability	Stable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/> Changing
End of Life	Permanent	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/> Soon	<input type="radio"/>
Robotic Performance					
Quality of Results	Adequate	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/> Inadequate	<input type="radio"/>
Number Bot Runners	Few	<input type="radio"/>	<input checked="" type="radio"/> Many	<input type="radio"/>	<input type="radio"/>
Task Characteristics					
Business Impact	Low	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/> High	<input type="radio"/>
Multi-System Robot?		Regulatory Compliance?			
No <input type="radio"/>		No <input type="radio"/>			

(b) Mapping of Input and Output Parameters

Process Variables	API Parameters
postalCode	plz
place	ort
district	ortsteil
street	strasse
streetNumber	hsnr
streetSuffix	hsnrZusatz
Output Variables	API Parameters
cplid	cplid

Fig. 7. Screenshots of Implementation Artefacts from the Prototype

7 Conclusion

We apply design science research to analyze how the implementation of RPA bots needs to be approached so that they can be easily replaced with APIs. Findings include six requirements to the design of migration-ready robots, eleven criteria with which the robots can be prioritized for transition, and one major challenge that requires particular attention. We organize these findings into a framework that integrates BPM and RPA from a methodological perspective. As the technological enabler, we provide an architectural blueprint that can be used to realize the application of the framework in an organizational setting. Finally, we demonstrate the migration towards APIs in a case study from practice.

However, this research is not without limitations. First, the sample size of the second qualitative evaluation is limited. Second, the prototype is implemented solely for illustration purposes, but cannot be brought into production without change. Third, the built artefacts have not been field-tested in an environment of actual use. Overcoming these limitations requires refinement of the framework for the development of migration-ready RPA bots via multiple experiences in multiple projects. Particularly, additional practitioners should be approached along further design cycles to ensure the results are both generally acceptable and complete. Future work could also include the conduction of field-tests so that the prototype is exposed to situations reflecting its intended use. This could uncover vulnerabilities in our architectural and conceptual considerations that may limit

³ For further illustration, the prototype is showcased in a video on [YouTube](#).

the practical applicability of the artefacts on an organizational and technological level. Nevertheless, the framework, the prototypical implementation, and their sample application introduce preparatory work that has not been explored so far. It provides the skeleton for future studies and presents new ground in closing the gap between frontend and backend automation.

References

1. van der Aalst, W.M.P.: Hybrid intelligence: to automate or not to automate, that is the question. *Int. J. Inf. Syst. Proj. Manag.* **9**(2), 5–20 (2021)
2. Axmann, B., Harmoko, H., Herm, L.-V., Janiesch, C.: A framework of cost drivers for robotic process automation projects. In: González Enríquez, J., Debois, S., Fettke, P., Plebani, P., van de Weerd, I., Weber, I. (eds.) *BPM 2021. LNBIP*, vol. 428, pp. 7–22. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85867-4_2
3. Brocke, J.v., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: *ECIS 2009 Proceedings*, pp. 2206–2217 (June 2009)
4. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*. Springer Berlin Heidelberg (2018). <https://doi.org/10.1007/978-3-642-33143-5>
5. Flechsig, C., Lohmer, J., Lasch, R.: Realizing the full potential of robotic process automation through a combination with BPM. In: Bierwirth, C., Kirschstein, T., Sackmann, D. (eds.) *Logistics Management. LNL*, pp. 104–119. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29821-0_8
6. Flechsig, C., Völker, M., Egger, C., Weske, M.: Towards an Integrated Platform for Business Process Management Systems and Robotic Process Automation, pp. 138–153 (Sep 2022)
7. Herm, L.-V., Janiesch, C., Reijers, H.A., Seubert, F.: From symbolic RPA to intelligent RPA: challenges for developing and operating intelligent software robots. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) *BPM 2021. LNCS*, vol. 12875, pp. 289–305. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_19
8. Hevner, A.R.: A three cycle view of design science research. *Scand. J. Inf. Syst.* **19**(2), 87–92 (2007)
9. 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
10. Kuckartz, U.: *Qualitative Inhaltsanalyse Methoden, Praxis. Computerunterstützung*. Beltz Juventa, WeinheimBasel (2016)
11. König, M., Bein, L., Nikaj, A., Weske, M.: Integrating robotic process automation into business process management. In: Asatiani, A., et al. (eds.) *BPM 2020. LNBIP*, vol. 393, pp. 132–146. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_9
12. Langmann, C., Turi, D.: *Robotic Process Automation (RPA) - Digitalisierung und Automatisierung von Prozessen*. Springer-Verlag GmbH (Sep 2021)
13. Myers, M.D., Newman, M.: The qualitative interview in IS research: examining the craft. *Inform. Organiz.* **17**(1), 2–26 (2007)

14. Plattfaut, R., Koch, J.F., Trampler, M., Coners, A.: PEPA: Entwicklung eines scoring-modells zur priorisierung von prozessen für eine automatisierung. *HMD Praxis der Wirtschaftsinformatik* **57**(6), 1111–1129 (2020)
15. Santos, F., Pereira, R., Vasconcelos, J.B.: Toward robotic process automation implementation: an end-to-end perspective. *Bus. Proc. Manag. J.* **26**(2), 405–420 (2019)
16. Smeets, M., Erhard, R., Kaußler, T.: Robotic process automation—background and introduction. In: *Robotic Process Automation (RPA) in the Financial Sector*, pp. 7–35. Springer, Wiesbaden (2021). https://doi.org/10.1007/978-3-658-32974-7_2
17. Wanner, J., Hofmann, A., Fischer, M., Imgrund, F., Janiesch, C., Geyer-Klingeberg, J.: Process selection in rpa projects - towards a quantifiable method of decision making. In: *International Conference on Information Systems 2019 (ICIS)*, pp. 1–17 (2019)
18. Weske, M.: *Business Process Management: Concepts, Languages. Architectures*. Springer, Berlin New York (2007). <https://doi.org/10.1007/978-3-662-59432-2>