

Research Challenges for Intelligent Robotic Process Automation

Simone Agostinelli, Andrea Marrella, and Massimo Mecella

Sapienza Università di Roma, Rome, Italy
{agostinelli,marrella,mecella}@diag.uniroma1.it

Abstract. Robotic Process Automation (RPA) is a fast-emerging automation technology in the field of Artificial Intelligence that allows organizations to automate high volume routines. RPA tools are able to capture the execution of such routines previously performed by a human user on the interface of a computer system, and then emulate their enactment in place of the user. In this paper, after an in-depth experimentation of the RPA tools available in the market, we developed a classification framework to categorize them on the basis of some key dimensions. Then, starting from this analysis, we derived four research challenges necessary to inject intelligence into current RPA technology.

1 Introduction

The recent developments in Artificial Intelligence (AI) force us to continuously revisit the debate on *what should be automated and what should be done by humans*. Robotic Process Automation [4] (RPA) is one of these developments. RPA is a fast-emerging automation approach that uses *software robots* (or simply *SW robots*) to mimic and replicate the execution of highly repetitive tasks performed by humans in their application's user interface (UI). SW robots are mainly used for automating office tasks in operations like accounting, billing and customer service. Typical tasks are: extract semi-structured data from documents, read and write from/to databases, copy and paste data across cells of a spreadsheet, open e-mails and attachments, fill in forms, make calculations, etc. [17].

The Gartner Hype Cycle for AI published in 2018¹ places RPA as one of the technologies at the peak of the hype cycle, meaning that there are nowadays deep expectations of what RPA will be able to deliver to the AI community. In addition, in recent years, a number of case studies have shown that RPA technology can concretely lead to improvements in efficiency for business processes involving routine work in large companies, such as O2 and Vodafone [5,12,10].

Despite this increased interest around RPA, when considering state-of-the-art RPA technology, it becomes apparent that the current generation of RPA tools is driven by predefined rules and manual configurations made by expert users rather than AI [13]. Starting from this statement, in this paper we identify and test ten RPA tools available in the market and categorize them by means

¹ <https://www.gartner.com/en/documents/3883863-hype-cycle-for-artificial-intelligence-2018>

of a classification framework. The results of the classification allow us to derive four research challenges required to evolve RPA towards AI.

2 Classification of RPA tools

Most of the actual deployments of RPA are industry-specific, e.g., for financial and business services [4]. According to [1], in 2019, the market of RPA solutions includes more than 50 vendors developing tools having different prices and features. Among them, we identified 10 vendors that offer to freely try their RPA tools, i.e., without the need to pay any license. The RPA tools in question are:

- *Automation Anywhere* (<https://www.automationanywhere.com/>)
- *AssistEdge* (<https://www.edgeverve.com/assistedge/>)
- *G1ANT* (<https://g1ant.com/>)
- *Kryon* (<https://www.kryonsystems.com/>)
- *Rapise* (<https://www.inflectra.com/Rapise/>)
- *TagUI* (<https://github.com/kelaberetiv/TagUI>)
- *UiPath* (<https://www.uipath.com/>)
- *VisualCron* (<https://www.visualcron.com/>)
- *WinAutomation* (<https://www.winautomation.com/>)
- *WorkFusion* (<https://www.workfusion.com/>)

We analyzed any of the above tools leveraging a dedicated case study based on a Purchase-to-Pay process, which includes many standardized and highly repetitive transactions with potential for automation [10]. After selecting the target process to automate, we employed the selected tools to design and train various SW robots, by recording the manual steps of the process. This has allowed us to identify a list of common tasks that must be performed to conduct a RPA project:

1. Determine which process steps (also called *routines*) are good candidates to be automated.
2. Model the selected routines in form of *flowchart diagrams*, which involve the specification of the actions, routing constructs (e.g., parallel and alternative branches), data flow, etc. that define the behaviour of a SW robot.
3. Record the mouse/key events that happen on the UI of the user’s computer system. This information is associated to the actions of a routine, enabling it to emulate the recorded human activities by means of a SW robot.
4. Develop each modeled routine by generating the software code required to concretely enact the associated SW robot on a target computer system.
5. Deploy of the SW robots in their environment to perform their actions.
6. Monitor the performance of SW robots to detect bottlenecks and exceptions.
7. Maintenance of the routines, which takes into account each SW robot’s performance and error cases. The outcomes of this phase enable a new analysis and design cycle to enhance the SW robots [11].

2.1 Classification Framework

We tested the selected RPA tools with our case study performing the tasks to conduct a RPA project. This has allowed us to realize a *classification framework* for RPA tools, which consists of the following key dimensions:

- **Software (SW) Architecture:** The specific SW architecture adopted by any tool: either *Stand-alone* or *Client-Server*.
- **Coding features:** The behaviour of SW robots can be defined with:
 - *Strong coding:* it is based on the realization of explicit programming scripts, often with the support of a command-line interface (CLI), that instructs the SW robots about the routines to emulate;
 - *Graphical User Interfaces (GUIs):* user friendly environments providing drag & drop facilities to build the flowchart of the routines to emulate;
 - *Low-code tools:* GUIs that – in addition to drag & drop facilities – provide low-coding functionalities to semi-automatically create software code.
- **Recording facilities:** The actions performed by a human within a software tool can be recorded in many ways:
 - *Web recording:* detection of user actions performed on a web browser;
 - *Desktop recording:* detection of user actions performed on a desktop UI;
 - *Others:* some RPA tools do not support neither web nor desktop recording. Nonetheless, they offer recording tools that work on specific applications only, such as Excel, Acrobat, SAP and Citrix. Some RPA tools provide also traditional screen-scraping recording.
- **Self Learning:** The ability of a RPA tool to automatically understand which user actions belong to which routines (*Intra-routine* learning), and which routines are good candidates for the automation (*Inter-routine* learning).
- **Automation type:** SW robots can either interact with users and/or acting independently. This leads to three different categories of automation:
 - *Attended:* the SW robots constantly require interaction with the users;
 - *Unattended:* the SW robots act like batch processes, i.e., manual intervention is not desired. This is ideal for optimizing back-office work;
 - *Hybrid:* Combination of the two above categories.
- **Routine composition:** The ability of a RPA tool to orchestrate through *manual support* or in *automated way* different (single) routines at run-time associated to different SW robots, when large workflows need to be emulated.
- **Log quality:** The quality of the logs recorded by RPA tools. Since routines consist of collection of activities to be enacted according to certain routing constraints, logs produced by RPA tools resemble *event logs* in process mining. To this end, we measure the quality of such logs using the classification provided in Process Mining Manifesto [2], where five maturity levels are defined, ranging from logs of excellent (★★★★★) to poor quality (★).

Table 1 shows the results of the application of our classification framework to the selected RPA tools. The following aspects become apparent: the majority of the tools provide (i) a Client-Server SW architecture, (ii) GUIs with drag & drop facilities and low-code functionalities, (iii) both web and desktop recording,

Table 1. Results of the application of the classification framework

Tool	SW Arch.		Coding			Recording			Self Learning		Autom. type	Routine comp.		Log quality
	Cl-Server	St.-alone	Strong	Low	GUI	Web	Desktop	Others	Intra-rout.	Inter-rout.		Manual	Autom.	
Aut. Anyw.	✓			✓	✓	✓	✓	✓			Hybrid	✓		★★★
AssistEdge		✓									Hybrid			★
G1ANT		✓	✓			✓	✓				Hybrid	✓		★★
Kryion	✓			✓	✓	✓	✓	✓			Hybrid	✓		★★
Rapise	✓			✓	✓	✓	✓	✓			Hybrid	✓		★
TagUI		✓	✓			✓	✓	✓			Hybrid	✓		★
UiPath	✓			✓	✓	✓	✓	✓			Hybrid	✓		★★
VisualCron	✓			✓	✓	✓	✓	✓			Attended	✓		★★★
WinAutom.	✓			✓	✓	✓	✓	✓			Hybrid	✓		★★
WorkFusion	✓			✓	✓	✓	✓	✓			Hybrid	✓		★★

(*iv*) a hybrid automation type, (*v*) manual-based features to achieve routine composition, (*vi*) logs of poor quality. Interestingly, differently from the other tools, G1ANT and TagUI offer strong-coding functionalities with a basic CLI to support the programming of SW robots. Finally, there is no tool that provides self learning or automated routine composition features.

3 Research Challenges

On the basis of the results discussed in the previous section, we have derived four research challenges (and potential approaches to tackle them) necessary to inject intelligence into current RPA technology.

1. **Intra-routine Self Learning (Segmentation).** Logs recorded by RPA tools are characterized by long sequences of actions and/or events that reflect a number of routine executions. A log can record information about several routines, whose actions and events are mixed in some order that reflects the particular order of their execution by the user [7]. In addition, the same routine can be spread across multiple logs, making the automated identification of routines far from being trivial. One possible approach to tackle this challenge is to rely on log analysis solutions in the Human-Computer Interaction field [9], which focus on identifying frequent user tasks inside logs consisting of actions at different granularity. Alternately, local process mining approaches [15] or sequential pattern mining [8] can be employed to identify sequential patterns of non-consecutive actions that tend to be repeated multiple times across multiple logs [7]. However, to date, no available solution exists that allows to automatically: (*i*) understanding which user actions have to be considered inside the log (separating noise to actions that contribute to routines); (*ii*) interpreting their semantics on the basis of their granularity and (*iii*) identifying to which routines they belong to.
2. **Inter-routine Self Learning (Automated identification of candidate routines to robotize).** To date, current RPA tools provide very limited support to this challenge, which is often performed by means of interviews, walkthroughs, direct observation of workers, and analysis of documentation that may be of poor quality and difficult to understand [11]. This manual

approach allows analysts to identify the most obvious routines, while it is not suitable to detect those routines that are not executed on a daily basis or that are performed across multiple business units in different ways [7]. Two potential solutions to this issue are provided respectively by [11] and [7], where the authors propose methods to improve the early stages of the RPA lifecycle using process mining techniques [3].

3. **Automated generation of flowcharts.** In RPA tools, there is a lacking of testing environments. As a consequence, SW robots are developed through a *trial-and-error* approach consisting of three steps that are repeated until success [16]: *(i)* First, a human designer produces a flowchart diagram that includes the actions to be performed by the SW robot on a target system; *(ii)* Second, SW robots are typically deployed in production environments, where they interact with information systems, with a high risk of errors due to inaccurate modeling of flowcharts; *(iii)* Third, if SW robots are not able to reproduce the behaviour of the users for a specific routine, then the designer adjusts the flowchart diagram to fix the identified gap. While this approach is effective to execute simple rules-based logic in situations where there is no room for interpretation, it becomes time-consuming and error-prone in presence of routines that are less predictable or require some level of human judgment. Indeed, the designer should have a global vision of all possible unfoldings of the routines to define the appropriate behaviours of the SW robot, which becomes complicated when the number of unfoldings increases. In cases where the rule set does not contain a suitable response for a specific situation, robots allow for escalation to a human supervisor. A possible solution to this challenge can be to resort on discovery algorithms from the process mining field [3] and to automatically extract flowcharts in form of Petri nets/BPMN models from RPA logs. Thus, it is necessary to investigate from the literature on process discovery [6] which algorithms suit better to extract the base structure of flowchart diagrams from a RPA log.
4. **Automated routines composition:** In modern contexts, human operators usually enact not just single tasks but complex workflows, consisting of many interrelated routines. However, current RPA technology allows to develop SW robots for executing single, independent routines. Only manual support is provided to orchestrate multiple routines, i.e., the management of more complex workflows is completely delegated to human supervisors. To synthesize complex execution strategies through an intelligent orchestration of the robots' routines, *automated planning* techniques in AI can be employed [14]. The idea is to consider the robots' routines like black boxes, i.e., as planning actions with specific preconditions and effects, and to delegate to a planning system the generation of a proper strategy to automatically compose them in a larger workflow that coordinates their orchestration.

It is worth to notice that, according to Table 1, the logs produced by the tested RPA tools have a poor quality (actions may be missing or not recorded properly), since they are mainly used for debugging purposes. Increasing the quality of RPA logs is a fundamental prerequisite to properly tackle the previous

challenges, which leverage a log analysis to discover, identify, model and compose routines in an automated way. To this end, RPA tools should aim at logs at the highest possible quality level.

To conclude, we note that our study has a threat to validity, since we analyzed only a sample of the RPA tools available in the market. As a consequence, our findings can not be generalized beyond the scope of the tested RPA tools. Nonetheless, we consider this work as an important first step towards the realization of intelligent solutions for RPA.

Acknowledgments. This research work has been partly supported by the “Dipartimento di Eccellenza” grant, the H2020 RISE project FIRST (grant #734599), the Sapienza grants IT-SHIRT, ROCKET and METRICS, the Lazio regional initiative “Centro di eccellenza DTC Lazio” and the project ARCA.

References

1. All 52 RPA Software Tools and Vendors: Sortable List [2019]. <https://blog.aimultiple.com/rpa-tools/> (2019)
2. van der Aalst, W., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., et al.: Process Mining Manifesto. In: BPM’11 Workshops (2011)
3. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg, 2 edn. (2016). <https://doi.org/10.1007/978-3-662-49851-4>
4. van der Aalst, W.M.P., Bichler, M., Heinzl, A.: Robotic Process Automation. BISE **60**(4) (2018)
5. Aguirre, S., Rodriguez, A.: Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. In: Applied Comp. Sc. in Eng. (2017)
6. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated Discovery of Process Models from Event Logs: Review and Benchmark. TKDE **31**(4) (2019)
7. Bosco, A., Augusto, A., Dumas, M., La Rosa, M., Fortino, G.: Discovering Automatable Routines From User Interaction Logs. In: BPM’19 Forum (2019)
8. Dong, G., Pei, J.: Sequence data mining, vol. 33. Springer Sc. & Bus. Media (2007)
9. Dumais, S., Jeffries, R., Russell, D.M., Tang, D., Teevan, J.: Understanding user behavior through log data and analysis. In: Ways of Know. in HCI (2014)
10. Geyer-Klingeberg, J., Nakladal, J., Baldauf, F., Veit, F.: Process Mining and Robotic Process Automation: A Perfect Match. In: BPM’18 Workshops (2018)
11. 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: CAiSE (2019)
12. Lacity, M., Willcocks, L.P., Craig, A.: RPA at Telefonica O2. Inst. Repo. for The London Sch. of Ec. and Pol. Sc. (2015)
13. Lohr, S.: The Beginning of a Wave: A.I. Tiptoes Into the Workplace. <https://www.nytimes.com/2018/08/05/technology/workplace-ai.html/> (2018)
14. Marrella, A.: Automated Planning for Business Process Management. J. Data Semantics **8**(2), 79–98 (2019). <https://doi.org/10.1007/s13740-018-0096-0>
15. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Mining local process models. Journal of Innovation in Digital Ecosystems **3**(2), 183–196 (2016)
16. Volodymyr, L., Dumas, M., Maggi, F.M., La Rosa, M.: Multi-perspective process model discovery for robotic process automation. In: CAiSE’18 Doct. Cons. (2018)
17. Willcocks, L.P., Lacity, M., Craig, A.: The IT function and robotic process automation. Inst. Repo. for The London Sch. of Ec. and Pol. Sc. (2015)