Ark Automate — an Open-Source Platform for Robotic Process Automation

Lukas **Hüller***[1]*, Kay Erik **Jenß***[1]*, Sandro **Speh***[1]*, Daniel **Woelki***[1]*, Maximilian **Völker***[1]* and Mathias **Weske***[1]*

[1]Hasso Plattner Institute, University of Potsdam, 14482 Potsdam, Germany

Abstract

With the recent uptake of Robotic Process Automation in research, also the tooling for building RPA bots comes into focus. So far, the mainly industry-driven domain produced several software programs for creating bots, and with this, different non-standardized modeling approaches emerged. Additionally, there are only few unrestricted or even open-source tools available. In this paper, we present Ark Automate, an open-source and web-based platform for creating and running RPA bots. It features interchangeable modeling interfaces, e.g., BPMN, to cater for the different user groups and is open for extension by the community.

Keywords

Robotic Process Automation, RPA Software, RPA Modeling

1. Introduction

By enabling companies to automate repetitive tasks on legacy systems with far less effort than traditional automation [1, 2], the adoption of Robotic Process Automation (RPA) is growing rapidly in industry. In the last few years, RPA has become a hot topic in research as well [3]. Despite the promising novel Robotic Process Mining approaches for bot generation, the manual design of RPA robots still plays an important role but has received much less attention in research.

The lack of a common RPA modeling standard led to a variety of proprietary modeling languages for RPA, trying to cater for the diverse user groups ranging from no-code users, to business users to IT experts. At the same time, there are only few open-source and platform-independent RPA tools, especially since Automagica was converted to closed source[1].

In this paper, we demonstrate a new type of RPA software, Ark Automate[2], which offers unique features for users as well as for the research community. It provides a graphical, web-based interface for modeling and also executing RPA bots and is built on top of the mature

[1]https://github.com/automagica/automagica/blob/234e14f/README.md

[2]Source code and installation guide: https://github.com/bptlab/ark_automate

Tutorial: https://github.com/bptlab/ark_automate/wiki/tutorial

Screencast: https://youtu.be/OLEIW9Ym9qw

RPA ecosystem of the Robot Framework[3]. Furthermore, the software is open source and can be extended, such as for evaluating different modeling languages for RPA, implementing robotic process mining approaches, or examining integrations with BPMS.

2. Features and Characteristics

Ark Automate is designed as a mainly web-based RPA software. Thus, all user interfaces, i.e., for modeling, starting, and supervising bots, are accessible to RPA developers and employees via the browser. The client software, responsible for the actual execution of bots, can be installed either locally or in virtual machines. Before describing the architecture in more detail in Sect. 3, some special features of Ark Automate are briefly introduced in the following.

2.1. Exchangeable Modeling Interface

To a large extent, RPA tools provide a visual modeling interface for creating RPA bots [3] to simplify the modeling process and make RPA accessible to business users as well. However, there is no standard modeling language for RPA yet, and consequently each RPA vendor has established its own format. But due to the broad circle of users, ranging from office employees, to process analysts to IT experts, it is unlikely that one interface will be able to cater for all needs, e.g., in terms of granularity or prior knowledge with modeling languages for workflows.

Therefore, Ark Automate is designed so that different modeling interfaces can be offered for creating bots[4]. To ensure collaboration among different user groups, the implementation allows users to switch between the different layouts at any time, thanks to a model-independent data structure that is used internally and updated in real time.

Currently, BPMN and a Robot code editor are supported, as conducted interviews revealed that BPMN is already widely used to document RPA bots and, moreover, is often already familiar to business users. The code editor, in contrast, is intended for technical users. New interfaces, such as flowcharts, can be added if required, e.g., for experimenting with new modeling methods for RPA.

2.2. Connection to BPMS

RPA is used for automating repeatedly executed tasks on computers [1]. However, these local tasks are often only small parts of whole business processes which are performed in a company [2]. To better integrate local RPA bots with potentially enterprise-wide processes, Ark Automate provides an application programming interface that can be used to control and orchestrate the execution of RPA bots.

This interface enables business process management systems (BPMS) to launch and monitor RPA bots at the right time in the process, allowing RPA bots to be seamlessly integrated into business processes.

---

[3]https://robotframework.org/

[4]However, many modeling languages are more powerful than the concepts offered by Robot Framework and thus need to be constrained. See https://github.com/bptlab/ark_automate/wiki/Documentation-Single-Source-of-Truth for more details.

2.3. Live Monitoring

Ark Automate enables the user and connected systems to closely monitor the current status of running bots in real time. It not only provides information about which operations have already been completed, are currently being processed, or are still to be executed, but it also supplies detailed error messages. This enables new possibilities for an improved exception handling in RPA, e.g., to reveal marginal cases that have not been considered at design time or to enable exception handling at the business process level.
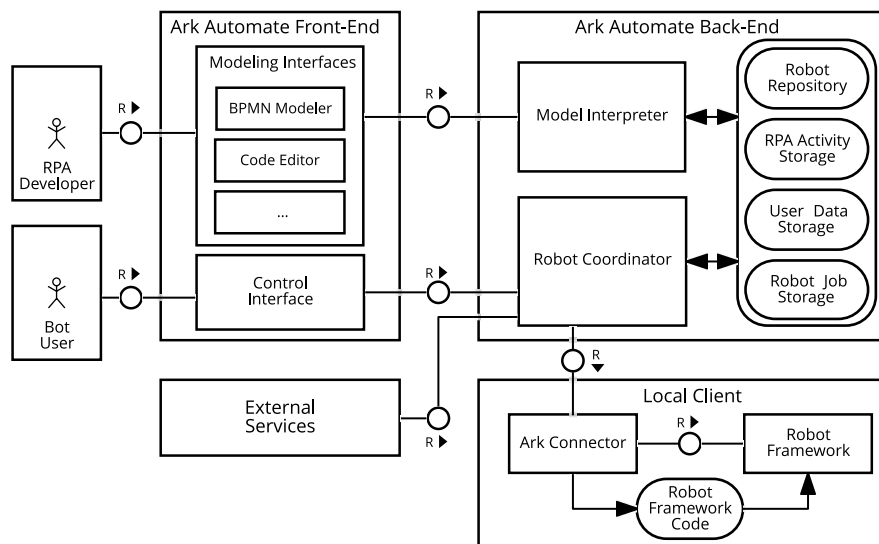
3. Architecture

The overall architecture of Ark Automate is given in Fig. 1.

To enable the functionality of different interfaces for modeling bots in the front-end, all bots are stored in the same model-independent format in the `Robot Repository`. In the back-end, the `Model Interpreter`, consisting of parsers for each interface, ensures that models can be translated between the internal representation and the respective modeling languages. General information and metadata about the various operations, i.e., building blocks available for creating a bot, are stored in the `RPA Activity Storage`.

The `Control Interface`, connected to the `Robot Coordinator` in the back-end, provides the functionality to start and configure new bot instances and closely monitor running executions (cf. Sect. 2.3). Additionally, the coordinator exposes an API for managing bot executions (see Sect. 2.2).

Created RPA bots can be executed at any computer having the `Local Client` installed. After startup, the client establishes a connection to the back-end and awaits new jobs. Whenever the execution of a bot is requested, the back-end retrieves the corresponding bot from the `Robot`

Figure 1: Architecture of Ark Automate including a front-end for modeling and managing bots, the back-end for interpreting models and coordinating bots, as well as the local client for bot execution

`Repository` and translates it from the internal data format to the actual Robot Framework code. This code is then transferred to the client software and executed using the `Robot Framework`. During execution, the client listens for updates from the framework and notifies the coordinator, which stores this information in the `Robot Job Storage`.

4. Example

One of the implemented interfaces supports the Business Process Management and Notation standard (BPMN). Figure 2 shows a screenshot of an RPA bot modeled using this interface. Each activity in the process can be assigned an RPA functionality using the sidebar on the right.

In this example, the activity "Enter Current Temperature in Cell" is configured to be a task for automating the *Excel.Application.* The opened drop-down then shows all operations available for this type of application, and the *Write to Cells* operation is selected in this case. Subsequently, input fields for configuring the selected operation appear below, here, for specifying the worksheet to modify, the column and row numbers, as well as the value to be entered, which in this case is a variable set by the previous "Get Current Temperature" activity. Each parameter can also be marked to be set at the start of the robot, instead of being hard-coded in the model. While designing the bot, the interface can be switched at any time, e.g., to view the modeled workflow in the actual Robot Framework code that is generated from the model and used for the execution of the bot.

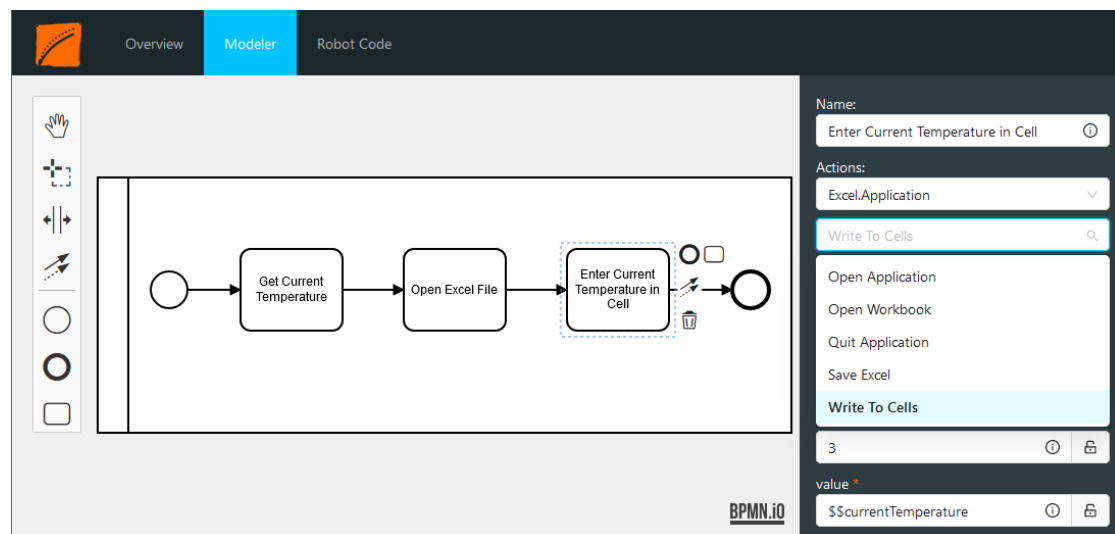After modeling, the bot can eventually be executed and the progress of execution can be observed live for each task.

Figure 2: RPA Bot modeled in the BPMN interface of Ark Automate

5. Maturity

Ark Automate is an open-source tool which was developed in collaboration with the project partner Camunda[5]. The key features and requirements for Ark Automate were derived from interviews with process experts, industry RPA developers, and researchers, revealing pain points of current RPA solutions as well as the needs of these different user groups, such as the discussed live monitoring and different modeling interfaces.

By using the open-source, text-based, and extendable Robot Framework, which is supported by the Robot Framework Foundation, a consortium of different companies, Ark Automate relies on a mature solution that is also used in industry to provide its RPA functionality.

6. Conclusion

In this paper, a novel open-source RPA tool, Ark Automate, was presented and offered to the research community for use and further development.

Focusing on the modeling of RPA bots, it encourages to reconsider the current state of RPA modeling. Building on the proven Robot Framework, the tool provides live monitoring of running bot instances, and thereby enables new research directions, especially regarding exception handling, a critical but neglected area in RPA [4]. Additionally, Ark Automate offers an API for a more profound integration with BPMSs, which should be further examined.

While the internal data structure already supports more advanced constructs, it is planned to reflect this complexity in the interface as well, e.g., by supporting branching with if/else or reusable bot sub-processes, and to evaluate more interfaces, like flowcharts. Furthermore, the live monitoring and exception handling aspects will be strengthened.

Acknowledgments

References

[1] W. M. P. van der Aalst, M. Bichler, A. Heinzl, Robotic Process Automation, Business & Information Systems Engineering 60 (2018) 269–272.

[2] C. Flechsig, J. Lohmer, R. Lasch, Realizing the Full Potential of Robotic Process Automation Through a Combination with BPM, in: Logistics management, Springer, 2019, pp. 104–119.

[3] J. G. Enriquez, A. Jimenez-Ramirez, F. J. Dominguez-Mayo, J. A. Garcia-Garcia, Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study, IEEE Access 8 (2020) 39113–39129.

[4] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. Leemans, C. Ouyang, A. H. ter Hofstede, I. van de Weerd, M. T. Wynn, H. A. Reijers, A. H. M. ter Hofstede, Robotic Process Automation: Contemporary themes and challenges, Computers in Industry 115 (2020).

---

[5]https://camunda.com/