



Conceptualizing Bots in Robotic Process Automation

Maximilian Völker^(✉) and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{maximilian.voelker,mathias.weske}@hpi.de

Abstract. Promising automation of repetitive tasks and release of manpower, Robotic Process Automation (RPA) continues to be a fast-growing market in the IT industry. The industry-driven advancement also comes with disadvantages. Each vendor established their own terminology and ecosystem, impeding communication, integration, and comparisons between RPA systems. In consequence, terminology and concepts are heterogeneous and not well understood. As a result, the scientific exchange lacks a consistent vocabulary. This paper proposes a vendor-independent conceptualization of RPA robots, their constituent parts and the relationships between those. It aims at providing a conceptual foundation for the further scientific investigation of RPA robots.

Keywords: Robotic process automation · Ontology · Conceptualization

1 Introduction

In times of unforeseeable developments in dynamic markets, we can witness the high value of creative human labor. As a result, organizations are facing the challenge to reduce the working time employees spend on repetitive and time-consuming tasks. With the electronic representation of business data, many data processing tasks are largely simplified. It might come as a surprise that the amount of human labor to process these data, actually, did not change much. Besides the shift from paper sheets to virtual files on computers, data still needs to be extracted from documents, transferred between systems and matched with other data sources.

Robotic Process Automation (RPA) promises to relieve workers from such repetitive tasks by offering a software solution that primarily simulates the interactions of a human user with the computer [2]. With the advancement of technology, activities that previously had to be performed manually can now be automated. Today, RPA is not only capable of simulating mouse clicks and keyboard interactions, but offers deep integration with many applications and, by using artificial intelligence techniques, acquires more and more human abilities, such as making decisions and processing unstructured data autonomously, which opens up entirely new use cases for automation [3, 5, 13, 19].

Triggered by its success in industry, RPA has only recently gained momentum in research. With missing conceptual underpinning of RPA and increasing complexity of the tools, the RPA vendors diverged in terms of concepts and terminology [13]. The lack of a shared understanding and conceptual foundation [9, 10, 17] hinders an overall comparison and overview, as well as the exchange of new ideas and approaches.

This paper aims at increasing the understanding of RPA by providing a conceptual basis for RPA bots, including their composition and their general capabilities. A vendor-independent ontology of RPA operations as well as the underlying concepts are introduced, categorized, and put into context. Furthermore, usage scenarios based on this conceptualization are presented, and its limitations are discussed.

The remainder of this paper is structured as follows. Section 2 presents related work from the field of robotic process automation. In Sect. 3, preliminary remarks about the underlying ontology are provided and basic RPA definitions are introduced. In Sect. 4, the conceptualization of RPA robots is presented, for which different usage scenarios are described in Sect. 5. The paper is concluded in Sect. 6, which includes points for future extension and research on this topic.

2 Related Work

Robotic Process Automation is increasingly attracting attention in research, which is expected to continue to grow in the upcoming years [10]. This technology utilizes “software agents called ‘bots’ that mimic the manual path taken by a human through a range of computer applications” [17]. However, there appears to be no generally accepted definition so far [10]. According to more traditional definitions, RPA is particularly suited for automating repetitive, well-structured, and rule-based workflows [1, 2, 11]. Other definitions, for example, also take capabilities of artificial intelligence into account, attributing RPA more flexible and complex characteristics [10, 17].

Current research in this area is aimed at different aspects, ranging from case studies (e.g., [2, 16]), to organizational considerations (e.g., [11]), to studying individual phases of the RPA lifecycle as introduced by Enriquez et al. [5]. However, there has been little focus on characterizing and classifying elementary concepts of RPA bots. Martínez-Rojas et al. [13] address a part of this problem by introducing a taxonomy of RPA components, i.e., building blocks, that rely on artificial intelligence. Motivated by the different naming schemes established by RPA software vendors, they propose a classification of such RPA components based on their capabilities using a unified terminology and link the corresponding vendor implementations to these concepts.

In this work, we do not focus on a specific technology such as AI in RPA, but want to provide a solid conceptual basis of RPA while abstracting from the concepts and the terminology used by specific vendors. For this purpose, a conceptualization of RPA bots is introduced.

3 Preliminaries

In the following, we introduce the foundations for the conceptualization of RPA bots presented in this paper and propose essential definitions for RPA.

Conceptualizations are often captured and formalized in ontologies to enable a common understanding of terms and concepts in the conceptualized domain [8]. In the domain of software, ontologies have a variety of applications, such as for identifying the nature of software errors [4] or for capturing the concepts and their interplay of software processes [6].

For the conceptualization presented in this work, the Core Software Ontology (CSO) introduced by Oberle et al. [14] will be used as a basis. This core ontology defines common, general terms in the software domain like `Data`, `Class`, and `Interface`. It is based on the foundational ontology DOLCE and some of its extensions such as the Ontology of Plans (OoP) and the Ontology of Information Objects (OIO) [7]. In the following, elements reused from this ontology are denoted by the prefix `CSO:`.

As discussed above, the term *RPA* is typically defined in a loose manner, often with different focal points. It is important to observe that RPA technology consists of several building blocks. This is similar to business process automation, with business processes, activities, and execution constraints as its components.

So far, the building blocks of RPA have rarely been defined or even identified explicitly, making it difficult to establish common ground for discussions and research. To overcome this problem, this section proposes definitions for the building blocks of RPA and outlines their respective characteristics. To continue the analogy with business process automation, RPA also has an equivalent of business processes—RPA bots (cf. Definition 1)—as well as activities—RPA operations (cf. Definition 2).

Based on the general definitions of RPA presented in the literature, the following definition of an RPA bot is introduced.

Definition 1 (RPA bot). An RPA bot is a machine-executable sequence of instructions consisting of RPA operations to automate interactions of human users with computer systems.

Definition 2 (RPA operation). An RPA operation is an atomic step in an RPA bot. It represents a single action performed on the system to be automated, usually, but not limited to, via cursor and keyboard interactions on the user interface level.

These RPA operations represent the smallest building blocks of RPA bots and are, to a large extent, already predefined by the respective RPA vendors.

Examples of RPA operations are manifold and range from internal operations of the RPA software, such as a variable assignment, to a mouse click on a button, to text extraction from documents. RPA operations should not be limited solely to interactions on the user interface, since RPA is capable of much more, such as performing database queries or calling web-services [9], and it is very likely that it will further evolve.

4 An Ontology of RPA Operations

Over the last few years, a number of RPA vendors have entered the market [1] and established their own terminologies [13]. However, most of the RPA functionality offered by these vendors does not differ much [9]. The conceptualization proposed in this paper captures and categorizes the concepts of RPA operations, i.e., the building blocks of RPA bots, in a vendor-independent manner. For building the ontology, the methodology introduced by Uschold and Gruninger [18] is applied.

4.1 Goal of Ontology and Competency Questions

The main goal of the proposed ontology is to provide a normative model [18] to create a shared understanding of RPA bots and RPA operations. Thus, reducing ambiguities and enabling researchers and practitioners to communicate without prior agreement on a vendor-specific terminology. The ontology is intended to provide a scaffold for later refinement and extension. This is important because RPA still evolves, such as through novel combinations with AI [13, 19].

The following informal competency questions (CQ) guided the development.

CQ 1: What is an RPA bot?

CQ 2: Which types of RPA operations exist?

CQ 3: Which types of software can be automated by a given RPA operation?

CQ 4: What is the needed input/output data for an RPA operation?

CQ 5: What kinds of operations can be applied to a certain file type?

CQ 6: What kinds of operations can be used to automate a given software?

CQ 7: Are there potential prerequisites for executing an RPA operation?

4.2 Ontology Capture

To capture the concepts in RPA, prominent and obvious terms are recorded first. Subsequently, concrete RPA operations, especially from the tools UiPath¹ and Automagica², are analyzed to further specialize and group the concepts.

Main Concepts. Besides the fundamental concepts of RPA bots and RPA operations defined in the previous section, data plays a crucial role for RPA (see also CQ 4 and 5). RPA bots are capable of extracting information from documents, they transfer data between different computer systems and can check and validate data [11, 17]. Another important facet to consider is the software that is automated using RPA (cf. CQ 3 and 6).

Both concepts, data and software, are already defined in the Core Software Ontology (CSO) and are reused here. `CSO:Software` is defined as a kind of `CSO:Data`, with the difference that software expresses a certain plan that defines

¹ <https://www.uipath.com/> (accessed 27.04.2021).

² <https://github.com/automagica/automagica/tree/ae8a1846f2> (latest open-source version, accessed 27.04.2021).

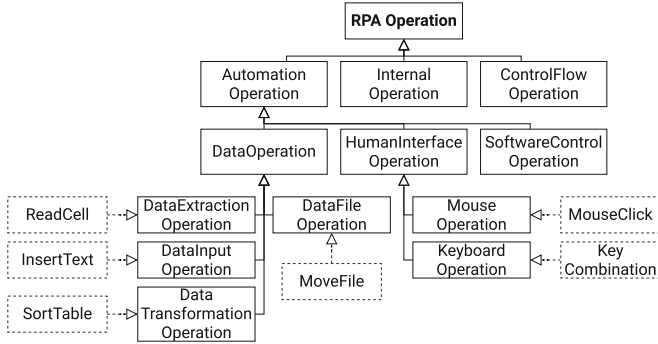


Fig. 1. Taxonomy of the concept `RPAOperation`. Terms framed with dashed lines are possible members of the concept and are not exhaustive.

a sequence of `CSO:ComputationalTasks` (cf. [14, 15]). We furthermore introduce `Files` for referring to `CSO:Data` that is not executable.

In the context of RPA, `RPABots` are a special kind of `CSO:Software`, which complies with the definition of RPA bots given in Definition 1, including the sequence of certain tasks or operations, respectively. Consequently, `RPAOperations` are a special kind of `CSO:ComputationalTasks` representing the smallest building blocks of software, just as `RPAOperations` are the building blocks of `RPABots`.

Specializations. To be able to answer more specific competency questions, such as CQ 2 or CQ 3, these main concepts are further subdivided.

In terms of software, there are different types that usually are automated using RPA. Besides the `RPABot` itself, the most prominent and obvious type are `Applications`, like browsers or office applications. They are the traditional target of `RPABots` and are usually automated by simulating mouse and keyboard inputs or, if available, via an API. Additionally, many RPA vendors allow the use of `Services` in `RPABots` to add a certain capability to the bot. A prominent example for a `Service` in RPA is optical character recognition (OCR) to retrieve text from images or documents. Also, basic operations on the `OperatingSystem` itself can be automated, like accessing the clipboard, stopping processes, or moving files.

To address CQ 2, operations are, as depicted in Fig. 1, further subdivided into `AutomationOperations`, `InternalOperations`, and `ControlFlowOperations`.

`ControlFlowOperations` are used to steer the sequence of operations within the bot, e.g., using *if/else* instructions or *loops*. `InternalOperations` do not access any data or software outside the bot, but mainly operate on internal variables, like matching a variable with a regular expression or adding a log message. `AutomationOperations` are the more apparent parts of a bot, they, for example, access external data, operate on applications, or call services.

`AutomationOperations` are further subdivided into three main concepts. `DataOperations` in general mark operations that access `CSO:Data` and are again

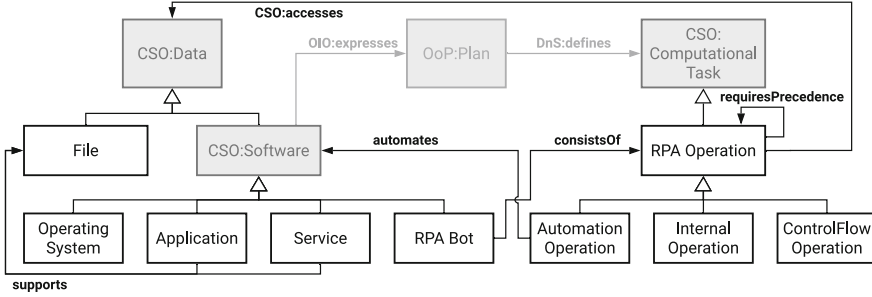


Fig. 2. Main elements of the ontology of RPAOperations

divided into different types (see Fig. 1), distinguishing whether data is read, written, transformed, or the file itself is manipulated. **HumanInterfaceOperations** identify operations that are usually independent of concrete **CSO:Software** and **CSO:Data**. They simulate the input of a human user, such as mouse clicks and key presses, and partly rely on so-called computer vision capabilities to locate elements on the UI. The third type, **SoftwareControlOperations**, label operations that manage software beyond the scope of data, i.e., they do not access data, but, for example, start or terminate **Applications** or **RPABots**.

Relations. Important relations between the concepts are introduced in the following. In general, **RPAOperations** can be used to automate a certain **CSO:Software** and to **CSO:access** **CSO:Data**.

To express how operations can be used to manipulate files, **CSO:access** is further specialized into **read**, **write**, and **transform** (the combination of the former two). These relations are a shortcut for the roles **CSO:Input** and **CSO:Output** in combination with the relations **CSO:inputFor** and **CSO:outputFor**, i.e., if an operation **reads** a certain kind of **CSO:Data**, this data plays the role of a **CSO:Input** for this operation (cf. [14, 15]). In the case of **DataOperations**, the type of operation already implies what kind of **CSO:access** takes places, e.g., **DataExtractionOperations** read **CSO:Data** (CQ 4), which is therefore an input for the respective operation.

The **automate** relation between **RPAOperations** and **CSO:Software** expresses that an operation utilizes or triggers a certain functionality offered by the software. While in some cases no distinct relation can be established, e.g., for generic UI operations, for software-specific operations, the **automate** relation can indicate for which type of software the operation can be used (CQ 3 and 6).

In most cases, applications and data are not independent as well. This connection is expressed by the **supports** relation, defining that an **Application** **supports** a certain **File** (can be inferred from **CSO:accesses** and **automates**).

Lastly, the relation **requiresPrecedence** is used to express constraints on the order of **RPAOperations** (cf. CQ 7), e.g., to express that the open operation needs to be executed before the application can be automated.

The described main concepts and their relations are depicted in Fig. 2.

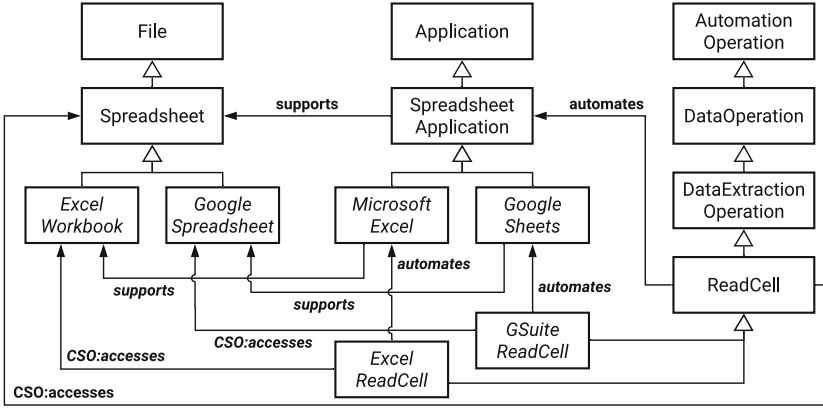


Fig. 3. Exemplary sub-concepts and *instances* related to the concept of reading a cell in a spreadsheet

Example. Figure 3 illustrates an example for the concept of the `RPAOperation` `ReadCell`, a `DataExtractionOperation`. Using the ontology, it can be expressed that this operation `CSO:accesses` (reads) data from `Spreadsheet` Files and that it can be used to `automate` `SpreadsheetApplication`s. A concrete instance realizing the concept of reading a cell in a spreadsheet is `ExcelReadCell`. It is related to the `Spreadsheet`'s instance `ExcelWorkbook`, which in turn is *supported* by the specific `SpreadsheetApplication` `MicrosoftExcel`.

4.3 Consideration of the Competency Questions

For evaluating ontologies, the research methodology proposes to review the competency questions in regard to the created ontology [18]. Table 1 lists the previously raised competency questions along with elements from the ontology that can be used to answer the respective question; it shows that all posed questions can be answered with the help of the concepts introduced.

Table 1. Review of the competency questions

CQ	Satisfied by
1 Nature of RPA bots	Type of <code>CSO:Software</code> that consistsOf <code>RPAOperations</code>
2 Types of operations	Taxonomy of RPA operations (see Fig. 1)
3 Automatable software	Taxonomy of software with <code>automates</code> relation
4 Input/output data	<code>CSO:accesses</code> with the types <code>read</code> , <code>write</code> , <code>transform</code>
5 Processable file types	Inverse of <code>CSO:accesses</code> relation
6 Operations for software	Inverse of <code>automates</code> relation
7 Prerequisites of operations	<code>requiresPrecedence</code> between RPA operations

5 Some Conceptualization Usage Scenarios

To illustrate the value of the RPA conceptualization introduced in this paper, this section outlines usage scenarios of the ontology and addresses its limitations.

Using the provided ontology, a knowledge base (KB) comprising available RPA operations can be built by including more operations and their relations like in the example (Sect. 4.2). Moreover, as the conceptualization captures real-world concepts observed across various RPA vendors, implementations by vendors can be mapped to the matching operations in the ontology.

5.1 Reducing the Maintenance and Change Effort

From time to time, it may be necessary to update RPA bots due to external changes or for improvements. So far, it required considerable effort to implement these changes, especially with the increasing number of automated software and RPA bots. Two conceivable scenarios should be highlighted here, the replacement of a software system automated using RPA and the change of the RPA vendor.

Replacing Software Systems. Continuing the example introduced in Fig. 3, we assume that *Microsoft Excel* is to be replaced by *Google Sheets*. In general, such a change makes any RPA bot inoperative, as the user interfaces differ.

By traversing the relations in the ontology, the required changes to the bots can, at least partly, be applied automatically. First, each operation used is checked whether it is affected by the change. In the example, the operation *ExcelReadCell* is linked to the affected software via the `automates` relation and thus must be replaced. Second, using the included taxonomies and relations, for each affected operation, the new appropriate operation, that realizes the same concept as before, must be identified. Here, the operation for reading a cell in *Google Sheets* must be found. By analyzing the `automates` relation of all instances of `ReadCell`, the operation *GSuiteReadCell* can be identified as the suitable replacement. Of course, this does not address any reconfiguration that may be required.

Changing an RPA Vendor. Provided that the aforementioned mapping of vendor-specific implementations to the knowledge base exists, the previously time-consuming and thus improbable switching of RPA providers can be facilitated. Figure 4 shows an example for such a mapping. Here, *ExcelReadCell*, an instance in the KB, is linked to matching implementations by vendors. The KB with the vendor mapping can automatically provide the appropriate, functionally equal operation of the new vendor to replace the old one. This approach not only allows to translate already implemented RPA bots between existing vendors, but also enables to first model bots independent of a vendor by using the

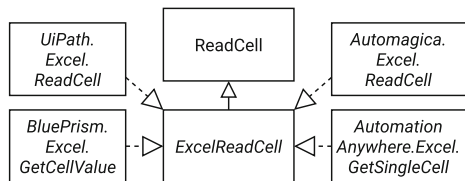


Fig. 4. Mapping of vendor implementations to an instance in the knowledge base

instances present in the KB which can later be translated to a specific vendor implementation using the mapping.

5.2 Enabling Automatic Documentation of RPA Bots

Interviews with RPA practitioners showed that despite the graphical modeling interfaces offered by most RPA software, RPA bots often have to be documented additionally. A reason for this could be that RPA operations are very narrow in scope, resulting in extensive models whose intent is difficult to grasp.

Using the introduced taxonomies, a simplified view on modeled RPA bots for documentation and communication purposes could be created. Assuming proper rules for abstraction, related, similar operations could be combined into an abstract activity. In addition, certain internal or preparatory operations could be removed in the documentation to further improve the clarity.

Moreover, the presented ontology could be used to replace vendor-specific, inexpressive labels by more informative and consistent names from the ontology.

Overall, the ontology and a resulting knowledge base could be used not only to (re)design bots, but also to enable an automated documentation, further reducing the overall implementation effort.

5.3 Limitations

This work focuses on RPA bots, their operations, and related software and data. Therefore, many other aspects of RPA are not covered here, such as management and methodological aspects, the execution or operating environment of bots, such as how exactly an application is automated, internal aspects like variables, or further configurations required for execution.

While the used core ontology already provides a formalized foundation, the degree of formality of the presented ontology could be further increased by using a more formal language to code the ontology and basing all relations on existing ones. This would be especially beneficial for a more automated processing, e.g., to realize the application ideas presented in this section.

6 Conclusion

This paper presents a conceptualization of RPA bots and their operations, revealing connections to data and applications that can be automated. Additionally, some potential applications were discussed, such as an abstracted, human-readable representation of bots, or the translation between different RPA vendors.

Furthermore, the ontology, especially when extended to a knowledge base, could prove useful in combination with existing approaches, like for providing possible labels for the analysis of textual process descriptions proposed by [12], or as an intermediate layer for implementations or research prototypes.

Besides, the ontology and its concepts could and should be extended as well. Here, several aspects are conceivable, like adding information on configurations of operations, i.e., more operational information, or adding information on the user interface elements automated by operations. This would eventually allow new RPA bots to be modeled using the knowledge base and later translated into vendor-specific bots with only minor adjustments, thus decoupling the modeling process from individual vendors.

References

1. van der Aalst, W.M.P., Bichler, M., Heinzl, A.: Robotic process automation. *Bus. Inf. Syst. Eng.* **60**(4), 269–272 (2018)
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. Chakraborti, T., et al.: From robotic process automation to intelligent process automation. In: Asatiani, A., et al. (eds.) BPM 2020. LNBIP, vol. 393, pp. 215–228. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_15
4. Duarte, B.B., Falbo, R.A., Guizzardi, G., Guizzardi, R.S.S., Souza, V.E.S.: Towards an ontology of software defects, errors and failures. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 349–362. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_25
5. Enriquez, J.G., Jimenez-Ramirez, A., Dominguez-Mayo, F.J., Garcia-Garcia, J.A.: Robotic process automation: a scientific and industrial systematic mapping study. *IEEE Access* **8**, 39113–39129 (2020)
6. Falbo, R.D.A., Bertollo, G.: A software process ontology as a common vocabulary about software processes. *IJBPM* **4**(4), 239–250 (2009)
7. Gangemi, A., Borgo, S., Catenacci, C., Lehmann, J.: Task taxonomies for knowledge content: Metokis deliverable D07 (2004)
8. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum. Comput. Stud.* **43**(5–6), 907–928 (1995)
9. Hofmann, P., Samp, C., Urbach, N.: Robotic process automation. *Electron. Mark.* **30**(1), 99–106 (2020)
10. Ivančić, L., Suša Vugec, D., Bosilj Vukšić, V.: Robotic process automation: systematic literature review. In: Di Ciccio, C., et al. (eds.) BPM 2019. LNBIP, vol. 361, pp. 280–295. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30429-4_19
11. Lacity, M.C., Willcocks, L.P.: A new approach to automating services. *MIT Sloan Manage. Rev.* Fall (2017). <http://eprints.lse.ac.uk/68135/>
12. Leopold, H., van der Aa, H., Reijers, H.A.: Identifying candidate tasks for robotic process automation in textual process descriptions. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 67–81. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_5
13. Martínez-Rojas, A., Barba, I., Enríquez, J.G.: Towards a taxonomy of cognitive RPA components. In: Asatiani, A., et al. (eds.) BPM 2020. LNBIP, vol. 393, pp. 161–175. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_11

14. Oberle, D., Grimm, S., Staab, S.: An ontology for software. In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*. IHIS, pp. 383–402. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3_17
15. Oberle, D., Lamparter, S., Grimm, S., Vrandečić, D., Staab, S., Gangemi, A.: Towards ontologies for formalizing modularization and communication in large software systems. *Appl. Ontol.* **1**(2), 163–202 (2006)
16. Penttinen, E., Kasslin, H., Asatiani, A.: How to choose between robotic process automation and back-end system automation? In: *ECIS 2018* (2018)
17. Syed, R., et al.: Robotic process automation: contemporary themes and challenges. *Comput. Ind.* **115**, 103162 (2020)
18. Uschold, M., Gruninger, M.: *Ontologies: principles, methods and applications*. *Knowl. Eng. Rev.* **11**(2), 93–136 (1996)
19. Viehhauser, J.: Is robotic process automation becoming intelligent? Early evidence of influences of artificial intelligence on robotic process automation. In: Asatiani, A., et al. (eds.) *BPM 2020. LNBIP*, vol. 393, pp. 101–115. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_7