

Towards a Taxonomy of Cognitive RPA Components

Antonio Martínez-Rojas, Irene Barba, and José González Enríquez^(✉)

Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática, Avenida Reina Mercedes, s/n, 41012 Sevilla, Spain
antonio.martinez@iwt2.org, {irenebr,jgenriquez}@us.es

Abstract. Robotic Process Automation (RPA) is a discipline that is increasingly growing hand in hand with Artificial Intelligence (AI) and Machine Learning enabling the so-called cognitive automation. In such context, the existing RPA platforms that include AI-based solutions classify their components, i.e. constituting part of a robot that performs a set of actions, in a way that seems to obey market or business decisions instead of common-sense rules. To be more precise, components that present similar functionality are identified with different names and grouped in different ways depending on the platform that provides the components. Therefore, the analysis of different cognitive RPA platforms to check their suitability for facing a specific need is typically a time-consuming and error-prone task. To overcome this problem and to provide users with support in the development of an RPA project, this paper proposes a method for the systematic construction of a taxonomy of cognitive RPA components. Moreover, such a method is applied over components that solve selected real-world use cases from the industry obtaining promising results .

Keywords: RPA · Artificial intelligence · Taxonomy

1 Introduction

The term Robotic Process Automation (RPA) refers to a software paradigm where robots are programs which mimic the behavior of human workers interacting with information systems (ISs) [17, 18, 26, 31], i.e. sets of components that perform actions that solve a particular RPA task. Such a paradigm has become increasingly popular due to RPA is of much interest to organizations. In such context, solutions that are based on Artificial Intelligence (AI)—called cognitive RPA [21] solutions—are receiving increasing attention since the combination of both disciplines offers several advantages. On the one hand, AI methods enhance RPA solutions by providing new capabilities. On the other hand, RPA solutions produce data regarding the own execution of the processes, that allows for improving the performance and accuracy of AI-based proposal, i.e., they enable a continuous training of the AI models. Therefore, main RPA platforms [20]

(e.g., BluePrism¹, UiPath², and Automation Anywhere³) already offer a battery of components that are based on AI techniques.

In the context of an RPA project, the RPA developer⁴ should take several decisions related to the robot design. Such decisions need to be based on the target RPA platform where robots are being developed. For this, the RPA developer needs to clearly understand such platforms, since one of the key factors which lead to the failure of RPA projects is the lack of understanding of these platforms [3]. This is especially important in the context of AI-based RPA solutions since they classify their AI-based RPA components in a way that seems to obey market or business decisions instead of common-sense rules. That is, components that present similar functionalities (e.g., “character recognition” is similar to “text language detection”) that are identified with different names and grouped within different categories depending on the platform that provides the components. For instance, UiPath platform considers that a task related to the recognition of a document element is classified as part of the group named *Document understanding*, while the BluePrism platform classifies such task within a group named *Document processing*. Therefore, analysing different RPA platforms to check their suitability for facing a specific need is typically a time-consuming and error-prone task.

This problem has been also pointed out by industry. To be more precise, the Servinform S.A. company, which considers AI-based RPA solutions as one of their strongest business lines, has identified a series of components that solve common use cases, in which the application of AI techniques is required. When developing these components, two main problems were found: (1) the task of selecting the most suitable platform is very challenging due to the heterogeneity of names and grouping of components, and (2) the task of training experts within the team to master one kind of component, such as *detecting elements in documents* or *natural language processing for conversations*, since these categories do not exist or are not easy to identify among the platforms. As a consequence, Servinform S.A. together with the IWT2 research group⁵ is currently involved in a research project, called AIRPA⁶, that is focused in the integration of AI techniques and RPA.

Figure 1 shows a graphical description of the motivation of this work. The RPA developer should decide how to design a robot to solve a cognitive task. For this, the developer has to analyze several AI-based RPA components that provide a solution to the problem. As can be observed, this component is identified with different names in each RPA platform which, moreover, present heterogeneous taxonomies. To provide support to RPA developers in the context of an

¹ <https://www.automationanywhere.com>.

² <https://www.uipath.com>.

³ <https://www.automationanywhere.com>.

⁴ <https://www.edureka.co/blog/rpa-developer-roles-and-responsibilities/>.

⁵ <http://iwt2.org>.

⁶ <https://www.servinform.es/wp-content/uploads/2020/02/Servinform-AIRPA-Publicidad-web-corporativa.pdf>.

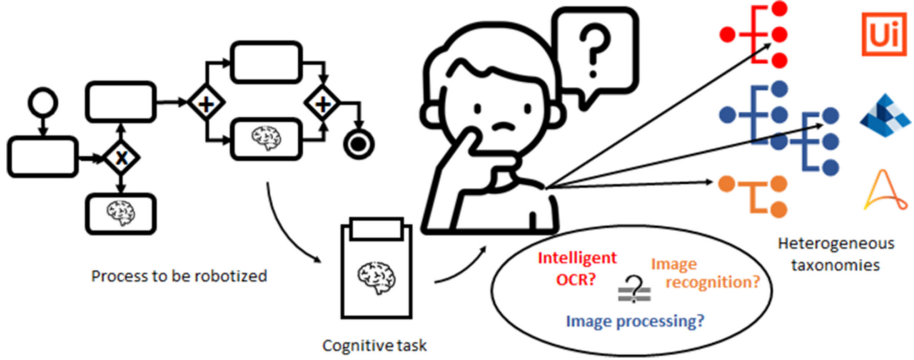


Fig. 1. Problem motivation

AIRPA project, this paper proposes a method for the systematic construction of a taxonomy of cognitive RPA components. This method is based on an incremental taxonomy which evolves a base taxonomy as needed. To be more precise, the application of the proposed approach leads to an initial taxonomy that can be extended and updated by following the incremental approach of the methodology. In addition, the proposed approach has been applied over a selection of RPA components that solve real-world use cases from industry. With such an application, it could be observed that the results that were obtained are very promising.

The rest of the paper is organized as follows. Section 2 describes the proposed method for the systematic construction of a taxonomy of cognitive RPA components. Section 3 presents the application example. Section 4 briefly summarizes related work. Finally, Sect. 5 concludes the paper and describes future work.

2 Systematic Construction of a Taxonomy of Cognitive RPA Components

To achieve a common classification for cognitive RPA components, named *AI-RPA taxonomy*, the following procedure has been carried out. Firstly, the available knowledge sources were identified. The sources that are considered are: (1) RPA platforms and organizations, and (2) human-knowledge that is provided by experts in such field. For example, in the application example described in Sect. 3, the sources are the UiPath, BluePrism and Automation Anywhere (hereafter AA) platforms, and experts from Servinform S.A. and IWT2 group.

The proposed approach considers a tree structure for the resulting taxonomy. In such a structure each node corresponds to a category. For defining the first level of the tree, taking as reference [28], a literature review on cognitive RPA taxonomies has been carried out. In such review, the works [9–11, 15, 24, 27] have been analyzed. It can be observed that [9] is the only work that provides an AI classification by type of application. It proposes a first level for the taxonomy

tree that covers all fields related to the application of AI to RPA. To be more precise, [9] proposes four categories: (1) classification, (2) skill acquisition, (3) continuous estimation, and (4) clustering. Considering these categories, a study was conducted to determine whether this classification could be used to group the studied AI-based RPA components. As a result, *classification* remains intact, *skill acquisition* and *continuous estimation* were adapted to *processing* and *governance* respectively, to bring their definition closer to RPA field. Meanwhile *clustering* is eliminated as a category, since it is considered as a technique used transversely in the other categories. This fact is justified by the very definition of the term given in this quotation “*Clustering is one of the most widely used techniques for exploratory data analysis. Across all disciplines, from social sciences to biology to computer science, people try to get a first intuition about their data by identifying meaningful groups among the data points*” [25]. Thus, it is an AI technique that will be the basis for the construction of components that are grouped in other categories, e.g. *classification* [19], but it cannot be defined as a category itself since this taxonomy does not group by technique but by application or functionality.

Thereafter, each of the categories was adapted to bring their definition closer to RPA field as described below.

- *Classification*: this term is used in both AI and RPA in the same way. As its name suggests, it comprises everything that encompasses a classification, from the traditional one by file type, to detection or recognition.
- *Processing*: includes functionality that requires skills acquisition, i.e., natural language *processing* [23] or intelligent image *processing* [7] to obtain a specific output.
- *Governance*: IT governance enables the effective use of IT which has a substantial impact on the value generated by IT investments [30,32]. In RPA, continuous estimation, similar to prediction and analysis—that is the basis of decision-making—are focused on process governance. The term governance is widely used in the field of RPA, as seen in [5,29]. One of the most representative examples of this is to determine which will be the next component to be used or if extra instances of a robot will be needed to cover the demand.

In the proposed approach, these categories will compose the first level of the AI-RPA taxonomy. In such taxonomy, when trying to classify a component, the aim will be to try to find the deepest possible category since the tree structure gives the taxonomy a hierarchical perspective. However, all the categories are defined by following the same procedure and have the same importance, including the ones that are placed in the first level.

This means that a component can be located in any of the nodes of the taxonomy, even if it is not a leaf node. This hierarchical structure will evolve dynamically, including new categories, for which these steps are followed⁷:

⁷ A similar procedure for classification was successfully applied previously in the context of Machine Learning knowledge [13].

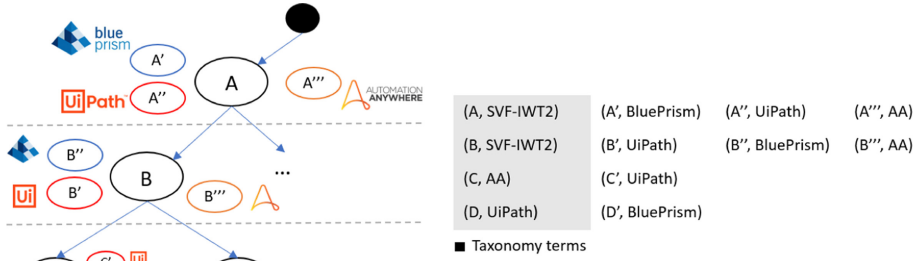


Fig. 2. AI-RPA unified knowledge store

1. A new term referring to an *AI – RPA* category is taken from one of the sources.
2. This term is compared to existing terms in the taxonomy. If the taxonomy is empty or there are no equivalent terms—a term that refers to the same category—, it is added as a child of the most similar category. Conversely, if an equivalent term is found, it is compared with the current category and, if it is considered that conveys the concept better than the current one, the latter is replaced.
3. Whenever a new term is added to the taxonomy, (1) the knowledge source from which it comes is saved (cf. Fig. 2), (2) the characteristics that make a component to belong to a category must be entered or updated, and (3) if it corresponds to a leaf of the tree that forms the taxonomy, the type of input that supports that category it is also indicated. For instance, if the term *Document Understanding* is added, it would be related to the term *Classification* → *Detection* → *Elements*, storing that it supports *Documents* as input and its origin lies in *UiPath*.

To better understand the structure of the *AI-RPA taxonomy*, each of its component elements is defined below.

Definition 1. An *AI-RPA taxonomy* $AIRPAT = (KnowlSources, CategoryTerms, TaxCats, CategoryChars)$ consists of

- *KnowlSources*: a set of tuples $\langle source_{id}, source_{name} \rangle$ which contains a unique id in the AIRPAT, and the name of the knowledge source.
- *CategoryTerms*: a set of tuples $\langle categoryTerm_{id}, categoryTerm_{name}, source_{id}, taxCategory_{id} \rangle$ which contains a unique id in the AIRPAT, the name of the category which is given in the knowledge source, an id of a knowledge source in *KnowlSources* which this category term comes from, and an id of a taxonomic category in *TaxCats*. The latter attribute aims to keep a synonymous relationship.
- *TaxCats*: a set of tuples $\langle taxCategory_{id}, taxCategory_{name}, parentTaxCateg_{id} \rangle$ which contains a unique id in the AIRPAT and a name of the taxonomic category, i.e., the category term that stands as the representative of the others.

- *CategoryChars*: a set of tuples $\langle categoryChars_{id}, categoryChars_{description}, taxCategory_{id} \rangle$ which contains a unique id in the AIRPAT, the description in $categoryChars_{description}$ of the characteristic that must be fulfilled by a component to belong to this category of the taxonomy, and an id of the taxonomic category in $TaxCats$.
- *InputFormatSupported*: a set of tuples $\langle inputFormSup_{id}, inputForm - Sup_{name}, taxCategory_{id} \rangle$ which contains a unique id in the AIRPAT, the taxonomic category to which this property points in $taxCategory_{id}$, and the name of the type of input that can support the taxonomic category in $inputForm.Sup_{name}$.

It is important to point out that each taxonomic category will have one or more characteristics or properties associated with it. For instance, for a single *TaxCateg*, there can be several *categoryChars* that define which characteristics a component must have to belong to it. Furthermore, it can support more than one input format, so it may have more than one *inputFormatSupported* associated. It can be deduced that in the case where no *TaxCateg* have been added to the taxonomy, only the first levels will be present. So a new category will always be added associated with one of the first levels, which are indispensable when initializing the taxonomy.

Thus, having defined all the elements to form the AI-RPA taxonomy, the incremental process to include new terms, described above in textual form, can now be defined in the form of an algorithm (cf. Algorithm 1).

Algorithm 1: How to increase the terms of the taxonomy

```

input : New CategoryTerm  $n$  , AI-RPA taxonomy  $at$ ,  $n$  origin source  $s$ 
if  $n$  not added to  $at$  then
  taxCategory  $c \leftarrow$  more similar category
  if  $s$  does not exist on  $at$  then
    | up  $s$  as  $(s_{id}, s_{name})$ 
  end
  if  $n$  better represents the concept than  $c$  then
    | up  $n$  as  $(n_{id}, n_{name}, s_{id}, c_{id})$ 
    | up  $c$  with  $(c_{id}, n_{name})$ 
  end
  if  $c$  is null then
    |  $p \leftarrow$  more suitable parent
    | up  $n$  as  $(n_{id}, n_{name}, s_{id}, p_{id})$ 
  end
end

```

In summary, using this taxonomy, the RPA developer will be able to find the component she needs or classify a given one according to its characteristics or *CategoryChars* (cf. Fig. 3). To find them, she only has to follow three steps: (1) go from the first level of the taxonomy downwards, checking which of the categories have characteristics that fit with the component ones; (2) filter the taxonomic categories whose characteristics are not fulfilled, and (3); go down to

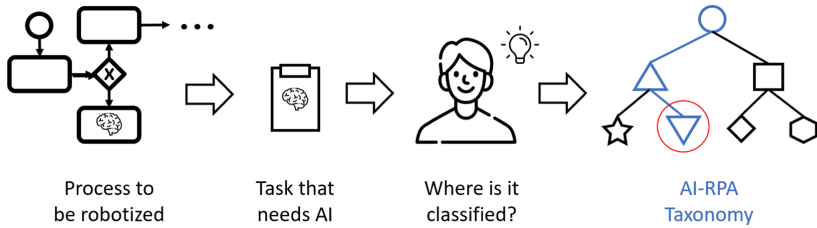


Fig. 3. Clear procedure to classify AI-RPA components

lower levels, doing the same with all the categories for each level, until reaching the deepest level of the tree. This way, the component to be found belongs to the categories that have not been filtered out after following these steps.

Thanks to the maintenance of traceability between the platforms and the terms of the taxonomy, it is also possible to automatically link each taxonomic category with the category in the own classification of each platform. Thus, the RPA developer will be able to find where the component that solves her problem is located, regardless of the platform the developer uses to build robots.

Consequently, seeking to unify cognitive RPA knowledge to facilitate this task to RPA developers, in the following section, it will be possible to see how the application of this methodology results in an initial and useful taxonomy. In addition, it can be extended due to the incremental nature of the proposed approach. Hence, the procedure described in this section can be further extended by performing successive iterations. Then, it is possible to perform a better classification of a great variety of cognitive RPA components.

3 Application Example

This section details the application of the proposed approach to selected real-world use cases from industry. Section 3.1 introduces the cognitive RPA platforms that have been selected. Section 3.2 describes the taxonomy that is obtained after applying the proposed approach to such selected platforms.

3.1 Selected Cognitive RPA Platforms from Industry

The cognitive RPA platforms that are selected are the following: UiPath⁸, Automation Anywhere⁹ and BluePrism¹⁰. We carefully reviewed the documentation of these platforms to obtain their cognitive categories and components, that are detailed in Table 1.

⁸ <https://docs.uipath.com/activities>.

⁹ <https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/using-the-workbench/cloud-commands-panel.html>.

¹⁰ <https://digitalexchange.blueprism.com/dx/search>.

Table 1. Heterogeneous taxonomies of selected cognitive RPA platforms

Platform	Category	Example of component
UiPath	UI automation (Computer vision)	CV Screen Scope
	Cognitive	Google Text translate or IBM Watson Text Analysis
	Document understanding	ML extractor
	Intelligent OCR (Document processing and PDF)	Intelligent form extractor
	ML services	ML Skill
	OCR	UiPath Screen OCR
BluePrism	Computer vision/Image processing	Tencent Cloud OCR
	Document processing	Elis Document Data Extraction
	Natural language processing	Natural Language Skill Google Cloud
	Expert Systems/Knowledge Base	Automated Fraud Investigation
	Machine learning	ML Engine Skill Google Cloud
	Workflow and decision engines	Appian Robotic Workforce Manager
	Visualization, monitoring reporting	ClearWork Process Orchestrator
	BI, Analytics and Big Data	Intelligent Decision Automation
Automation Anywhere	Conversational AI/Bots/Virtual Agents	Human-Robot Conversations
	There are no categories, it offers directly a list of components named “packages”	Fuzzy match
		IBM Watson Speech to Text
		Image recognition
		Microsoft LUIS NLP
		OCR
		IQ Bot

For instance, considering an intelligent document processing problem, UiPath classifies it as *Document understanding* or *Intelligent OCR*, BluePrism as *Document Processing* and Automation Anywhere as *IQ Bot* [14]. Some even refer directly to the name of the component instead of categorizing it. For example, Automation Anywhere, where these do not belong to any category (*Microsoft LUIS*) or Speech-Text (*IBM Watson Speech*). As can be observed in Table 1, the taxonomies of the different cognitive RPA platforms that were analyzed are heterogeneous. Therefore, in this scenario, the application of the proposed approach is desired to obtain a homogeneous taxonomy of cognitive RPA components.

3.2 Resulting Taxonomy

The proposed taxonomy (cf. Fig. 4) is put into practice considering the knowledge of Servinform and IWT2 and the review performed in Sect. 2. The taxonomy follows a tree structure, whose nodes represent the taxonomic categories and the black circles attached to the leaf nodes correspond to the type of input they support. Even though, only a part of it is shown in the category tree, is composed

of four parts that give content and accumulate the knowledge hidden behind the nodes.

1. The first one composed of the terms *classification*, *processing*, and *governance*. Firstly, these terms will be instantiated as the first step in defining any *AI – RPA* taxonomy. Their objective is to cover all areas of application of AI in the RPA, as well as carrying out pruning to achieve a quick classification.
2. The second part is formed by the rest of the nodes, which are more specific categories, hanging from the nodes of the first level. These categories may be modified or even increased being able to have as many children as terms can be included from available knowledge resources.
3. The third part corresponds to the maintenance of traceability of the terms added in the second part. Thus, the equivalent terms to each taxonomic category and the knowledge sources from which they come will be stored (cf. Fig. 2).
4. Finally, the fourth part is formed by the black-colored properties, corresponding to the input format supported by the category. These nodes do not belong to the tree structure, but they describe properties that allow differentiating the components according to the supported input type. For example, it can be distinguished between components that take *Text*, *Documents* or *Audio* as input. Note that these nodes will be conditioned by the category to which it belongs. For example, the *Image* option will not be included for a *Translation* category.

Note, that, since the information available to determine the belonging to the categories in the main platforms is minimal, the *CategChars* of each *TaxCateg* are being obtained as an effort from both *Servinform* and *IWT2*.

Hence, each of the *TaxCategs* presented in Fig. 4 are listed in order from left to right, in which the *CategoryChars* correspond to the items under each of them.

1. Classification

- It takes as input a list of classes.
- It takes a set of elements as input.
- It finds association between classes/categories and elements.

I Detection

- It takes an input from which a specific classification is extracted.
- The input can be a file from which it is necessary to deduce to which specific class it belongs or to identify the entities within it that comply with a specific classification.
 - i. Elements
 - It extracts the elements that meet specific characteristics that have been taken as input.
 - ii. Anomalies
 - Characteristics of a non-anomalous element as input.

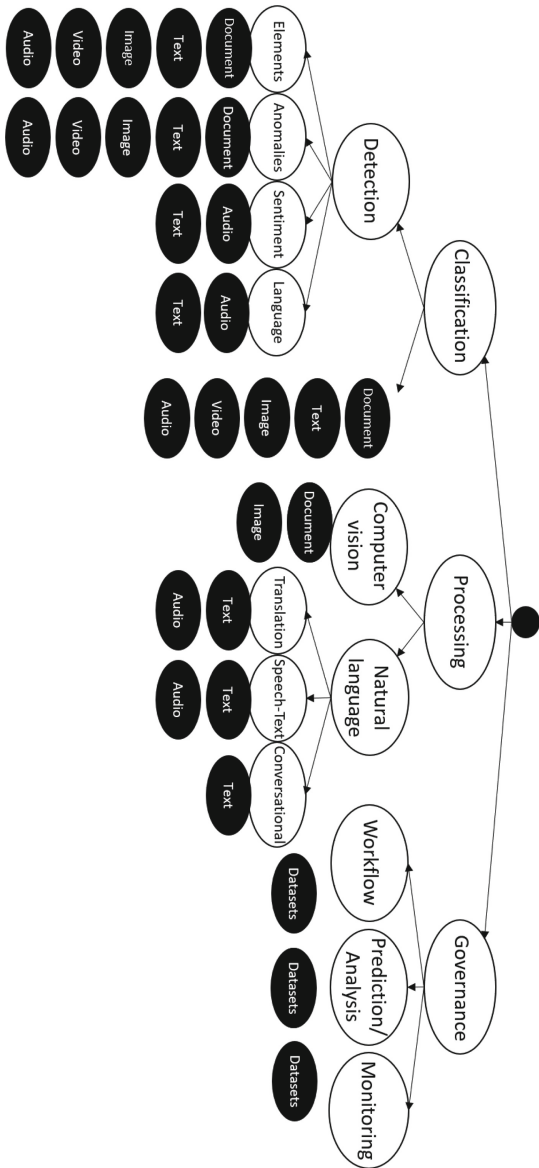


Fig. 4. Resulting taxonomy

- Of the elements it takes in the entry, it performs a binary classification between two classes, yes/no anomalous.
- iii. Sentiment
 - It takes as input audio or text.

- It takes as classes a list of possible feelings, and gets as output one of them.
- iv. Language
- It takes as input audio or text.
 - It takes as classes the existing languages and gets as output one of them.
- 2. Processing**
- The output is obtained by transforming or modifying the input.
- I Computer vision
- It takes as input an image or a document.
 - It extracts from the input, concrete information that is visually inferred.
- II Natural language
- It takes as input an image or a document.
 - It transforms the input according to some of its characteristics, such as language or format, or makes an interpretation of the input to obtain a coherent output to it.
- i. Translation
- As its name suggests, performs an interpretation of its input to translate it into a specified language.
- ii. Speech-Text
- It extracts the message contained in the entry, and transforms it into another format.
 - If it is an audio input, it transforms the input into text.
 - If it is a text input, it transforms it into audio.
- iii. Conversational
- It takes an input from interaction with a human through language, either written or spoken.
 - According to the input, it interprets its meaning and generates as output a coherent response.
- 3. Governance**
- It takes as input a set of data concerning a decision
 - The outputs obtained are aimed at optimizing.
 - It is on a higher level than other instances or components, so that it carries out control over them.
- I Workflow
- The component takes as input the data related to the process, such as the result of the last action.
 - The component determines which is the best action to take next according to the parameters.
- II Prediction/Analysis
- It takes as input the historical and/or environmental data of the prediction.
 - It makes a forecast or regression of what is going to happen.
- III Monitoring

- It carries out a real-time control of the process.
- It is waiting for a failure to appear during the execution of the process when this happens it decides resolve it.

As mentioned above, for this taxonomy, the classification of a given component would be as simple as checking which of the characteristics it meets. It is important to note that it may belong to several of the taxonomic categories.

Making it easier for the RPA developer to find a component, simply by continuing to check which of the categories (*TaxCateg*) cover the characteristics of its problem (cf. Fig. 3). In this way, following the procedure defined in Sect. 2, RPA developers will be able to find out exactly which categories a component corresponds to.

4 Related Work

Some previous works related to addressing the problem of classifying AI techniques and RPA components have been found. This is the case of the taxonomies or classifications proposed for AI according to their application, the learning paradigm, or the algorithm used [6, 9, 11, 12, 15, 22, 24, 27, 33].

Even though all the work present taxonomies applied to different areas of AI or automation, the only one in which a taxonomy explicitly aimed at RPA is proposed is in this [8]. Two proposes [1, 2] that present specific taxonomies of cognitive RPA components are the closest ones related to the approach presented in this paper. However, unlike the proposed approach, they do not allow the dynamic generation of taxonomies.

Initial proposals for taxonomies can also be found in papers from more specific fields [4, 16]. In such context, all proposals follow the same form of definition, i.e., a classification that is not iterative, incremental nor extensible. Initial proposals for taxonomies can also be seen both in the papers mentioned above and those from other specific fields as [4, 16]. However, all proposals follow the same form of definition, i.e., again an unchanging classification is proposed that is not iterative, incremental or extensible.

To summarize, although the works [1, 2] are close to the topic of this paper, to the best of our knowledge, any previous work proposes an incremental classification for the cognitive components in RPA. And this is where the need identified by [28] comes up. “*There is a strong interest in taxonomies in Software Engineering, but few taxonomies are extended or revised. Taxonomy design decisions regarding the used classification structures, procedures and descriptive bases are usually not well described and motivated*”. This need is addressed in the proposed approach since it does not only proposes a classification for cognitive RPA components, but an extensible taxonomy is proposed following a defined procedure. Hence, such taxonomy can be updated and incrementally extended when necessary. Note that for the development of this work, the resources available in the literature related to the terms *robotic process automation*, *taxonomy*, *cognitive*, *machine learning* have been reviewed and only the papers that are listed in the references have been found.

5 Conclusions and Future Work

In the context of AI-based RPA solutions, the manual design of cognitive tasks is currently a time-consuming and error-prone task due to the heterogeneity that is given in the names and classifications of the different RPA platforms. To support RPA developers in this task, this work presents an approach for the systematic construction of a taxonomy of cognitive RPA components that are offered by different AI-based RPA platforms. Such approach is framed within a research project that is the result of a collaboration between the Servinform S.A. company and the IWT2 research group. The proposed approach is applied over selected components that solve real-world use cases from industry, and very promising results are obtained.

Unlike previous related work (e.g., [1,2]), the proposed approach does not propose a specific taxonomy but a method for systematically generating such taxonomy from the information that is provided by the different RPA platforms. Therefore, the taxonomy can be generated as many times as required, resulting in a dynamic process in which the resulting taxonomy can be extended and updated when necessary. Note that this is a great added value since the cognitive RPA market is growing by leaps and bounds. Furthermore, unlike previous related work, the proposed approach is focused on specific RPA platforms, i.e., on platforms that provide AI-based solutions.

For future work, we intend to consider the specification of the characteristics of the categories of the cognitive taxonomies as defined rules, to be able to use them for objective classification by a software system. In addition, an automatic classification of cognitive tasks in the resulting taxonomy according to using AI techniques is intended to be analyzed.

Acknowledgements. This research has been supported by the Pololas project (TIN2016-76956-C3-2-R) of the Spanish Ministry of Economy and Competitiveness, the Trop@ project (CEI-12-TIC021) of the Junta de Andalucía, and the AIRPA (P011-19/E09) project of the Centro para el Desarrollo Tecnológico Industrial (CDTI) of Spain.

References

1. IEEE guide for terms and concepts in intelligent process automation. IEEE Std 2755-2017, pp. 1-16 (2017)
2. IEEE guide for taxonomy for intelligent process automation product features and functionality. IEEE Std 2755.1-2019, pp. 1-53 (2019)
3. ABBYY. State of Process Mining and Robotic Process Automation 2020 (2020). www.abbyy.com/en-us/solutions/process-intelligence/research-report-2020. Last Accessed May 2020
4. Alaydie, N., Reddy, C.K., Fotouhi, F.: Exploiting label dependency for hierarchical multi-label classification. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012. LNCS (LNAI), vol. 7301, pp. 294-305. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30217-6_25

5. Teemu, P.E.A., Aleksandre, K.: Unexpected problems associated with the federated it governance structure in robotic process automation (RPA) deployment. D4 julkaistu kehittämis- tai tutkimusraportti tai -selvitys (2019)
6. Baltrušaitis, T., Ahuja, C., Morency, L.-P.: Multimodal machine learning: a survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(2), 423–443 (2018)
7. Batchelor, B.G.: *Intelligent Image Processing in Prolog*. Springer Science & Business Media, London (2012). <https://doi.org/10.1007/978-1-4471-0401-8>
8. Beerbaum, D.: Artificial intelligence ethics taxonomy-robotic process automation (RPA) as business case (2020)
9. Golstein, B.: SharperAI, CEO. A Brief Taxonomy of AI (2018). <https://www.sharper.ai/taxonomy-ai/>. Last Accessed May 2020
10. Bkassiny, M., Li, Y., Jayaweera, S.K.: A survey on machine-learning techniques in cognitive radios. *IEEE Commun. Surv. Tutor.* **15**(3), 1136–1159 (2013)
11. Davis, J., Hoffert, J., Vanlandingham, E.: A taxonomy of artificial intelligence approaches for adaptive distributed real-time embedded systems. In: 2016 IEEE International Conference on Electro Information Technology (EIT), pp. 0233–0238. IEEE (2016)
12. Ding, R.-X., et al.: Large-scale decision-making: characterization, taxonomy, challenges and future directions from an artificial intelligence and applications perspective. *Inf. Fusion* **59**, 84–102 (2020)
13. Enriquez, J.G., Martínez-Rojas, A., Lizcano, D., Jiménez-Ramírez, A.: A unified model representation of machine learning knowledge. *J. Web Eng.* **19**, 319–340 (2020)
14. Everest Group Research. Everest Group PEAK Matrix™ for Intelligent Document Processing (IDP) Technology Vendors 2020, March 2020
15. Feldt, R., de Oliveira Neto, F.G., Torkar, R.: Ways of applying artificial intelligence in software engineering. In: 2018 IEEE/ACM 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), pp. 35–41 (2018)
16. Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., Herrera, F.: Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study. *IEEE Trans. Evol. Comput.* **14**, 913–941 (2010)
17. Fung, H.P.: Criteria, use cases and effects of information technology process automation (ITPA). *Adv. Robot. Autom.* **3**, 1–10 (2014)
18. 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
19. Larson, R.R.: Classification clustering, probabilistic information retrieval, and the online catalog. *Libr. Quart.* **61**(2), 133–173 (1991)
20. Le Clair, C., O'Donnell, G., Lipson, A., Lynch, D.: The forrester wave™: robotic process automation, Q4 2019. *The Forrester Wave* (2019)
21. NASSCOM. Cognitive RPA, The Future of Automation (2019). <https://www.nasscom.in/knowledge-center/publications/cognitive-rpa-future-automation>. Last Accessed May 2020
22. Olsson, H.H., Crnkovic, I.: A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In: Agile Processes in Software Engineering and Extreme Programming: 20th International Conference Canada, 21–25 May 2019, Proceedings (2019)
23. Rainey, S.K., Brown, B., Kirk, D.B.: Bots, natural language processing, and machine learning. *Tax Executive* **69**, 39 (2017)

24. Ramírez-Gallego, S., García, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Alonso-Betanzos, A., Manuel Benítez, J., Herrera, F.: Data discretization: taxonomy and big data challenge. *Wiley Interdisc. Rev. Data Min. Knowl. Discov.* **6**(1), 5–21 (2016)
25. Shalev-Shwartz, S., Ben-David, S.: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York (2014)
26. Slaby, J.R.: Robotic automation emerges as a threat to traditional low-cost outsourcing. *HfS Res. Ltd* **1**(1), 3 (2012)
27. Unterkalmsteiner, M., Feldt, R., Gorschek, T.: A taxonomy for requirements engineering and software test alignment. *ACM Trans. Softw. Eng. Method. (TOSEM)* **23**(2), 1–38 (2014)
28. Usman, M., Britto, R., Börstler, J., Mendes, E.: Taxonomies in software engineering: a systematic mapping study and a revised taxonomy development method. *Inf. Softw. Technol.* **85**, 43–59 (2017)
29. Vasarhelyi, M.A.: Formalization of standards, automation, robots, and IT Governance. *J. Inf. Syst.* **27**(1), 1–11 (2013)
30. Weill, P., Ross, J.: A matrixed approach to designing it governance. *MIT Sloan Manage. Rev.* **46**(2), 26 (2005)
31. Willcocks, L., Lacity, M., Craig, A.: Robotic process automation: strategic transformation lever for global business services? *J. Inf. Technol. Teach. Cases* **7**(1), 17–28 (2017)
32. Wu, S.P.J., Straub, D.W., Liang, T.P.: How information technology governance mechanisms and strategic alignment influence organizational performance: insights from a matched survey of business and it managers. *MIS Q.* **39**(2), 497–518 (2015)
33. Zaidan, A.A., Zaidan, B.B.: A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations. *Artif. Intell. Rev.* **53**(1), 141–165 (2020)