Knowledge and Information Systems (2023) 65:3633–3664
https://doi.org/10.1007/s10115-023-01876-9

REGULAR PAPER

Automatic generation of incremental taxonomies for supporting the users in the development of an RPA project

Antonio Martínez-Rojas[1] · Irene Barba[1] · Carmelo Del Valle[1] · Andrés Jiménez-Ramírez[1] · José González-Enríquez[1]

Received: 17 January 2022 / Revised: 31 March 2023 / Accepted: 1 April 2023 /
Published online: 28 April 2023

Abstract

The robotic process automation (RPA) paradigm is a discipline that is becoming increasingly popular thanks to the great interest shown by the industry. In such context, RPA solutions based on artificial intelligence, i.e., cognitive solutions, are receiving increasing attention. In a cognitive RPA project, the RPA developer is in charge of selecting the most suitable components that solve specific tasks from the sets of components provided by different RPA platforms. This selection is very challenging, especially since there is no homogeneity in component names or component classifications. Such a situation turns an RPA project's development into a time-consuming, error-prone, and very tedious process. Therefore, supporting the RPA developer in developing a cognitive RPA project is desired. The industry has also pointed out this need. This work presents a proposal for supporting the users in developing a cognitive RPA project. To be more precise, an incremental method to automatically generate taxonomies from cognitive RPA platforms is proposed. Such taxonomies can be dynamically adapted when necessary. In previous work, the initial aspects of this research were presented. However, the current work greatly enhances such previous work by: (1) extending the proposed method to improve the management of real-world use cases from industry, (2) developing a proof-of-concept tool that is based on the proposed approach, (3) validating the proposed method by applying it to real-world use cases from industry, and (4) performing a literature review on related topics. The results obtained are auspicious and demonstrate that the proposed approach substantially improves the support given to users during the development of a cognitive RPA project.

Keywords Robotic process automation · Artificial intelligence · Taxonomy

✉ Antonio Martínez-Rojas
amrojas@us.es

[1] Department of Computer Languages and Systems, University of Seville, Sevilla, Spain

1 Introduction

The term robotic process automation (RPA) refers to a software paradigm where robots are software that mimics the behavior of human workers interacting with information systems [14, 16, 25, 28]. These robots can be defined as a set of components developed to automate a particular block of tasks in a business process (BP) context. For instance, the receipt of orders represents a block of tasks within the management of orders. This block contains tasks such as *extracting incoming orders via email* or *registering orders in an Enterprise Resource Planning (ERP) system* that are solved by using different components.

The RPA paradigm is becoming increasingly popular thanks to the great interest shown by the industry, especially by organizations that have traditionally adapted to new technologies quickly, such as process-aware information systems [19, 27].

RPA solutions based on artificial intelligence (AI)—cognitive RPA solutions [22] henceforth—are receiving increasing attention due to the significant advantages of integrating AI techniques in the field of RPA. Note that, on the one hand, AI techniques allow for the automation of tasks traditionally carried out manually. Moreover, on the other hand, RPA solutions produce a large amount of data during their execution that can be stored and used for continuous training of AI models, which leads to increasing both the accuracy and the performance of cognitive RPA solutions.

The increasing presence of cognitive RPA solutions in the market can be observed in the growing number of cognitive components provided by the main RPA platforms, such as BluePrism,[1] UiPath,[2] or Automation Anywhere[3] [20]. Moreover, many RPA vendors and companies (such as Servinform S.A.[4]) consider cognitive RPA solutions as one of their most robust business lines [5].

1.1 Motivation

In the context of a cognitive RPA project, the RPA developer is in charge of designing the process, defining the automation architecture, and managing the production.[5] Thus, the RPA developer needs to select the most suitable components that solve specific tasks from the sets of components that different RPA vendors provide.

This selection is very challenging, mainly since there is currently no homogeneity in component names or component classifications. More precisely, nowadays, components with almost identical functionality are named differently and classified into different categories depending on the RPA platform they are provided (cf. Fig. 1).

This situation turns the development of an RPA project into a time-consuming, error-prone, and very tedious process in which both the satisfaction of the RPA developer as well as the correctness and efficiency of the resulting software can be seriously compromised. Therefore, supporting the RPA developer in the development of a cognitive RPA project is desired. The industry has also pointed out this need. Specifically, the Servinform S.A. company carried out a comprehensive analysis of the process of developing a cognitive RPA project and, based on their know-how and practical experience, concluded that: (1) the heterogeneity in names

[1] Available at: https://www.blueprism.com.

[2] Available at: https://www.uipath.com.

[3] Available at: https://www.automationanywhere.com.

[4] Available at: https://www.servinform.es/.

[5] Available at: https://www.edureka.co/blog/rpa-developer-roles-and-responsibilities/.

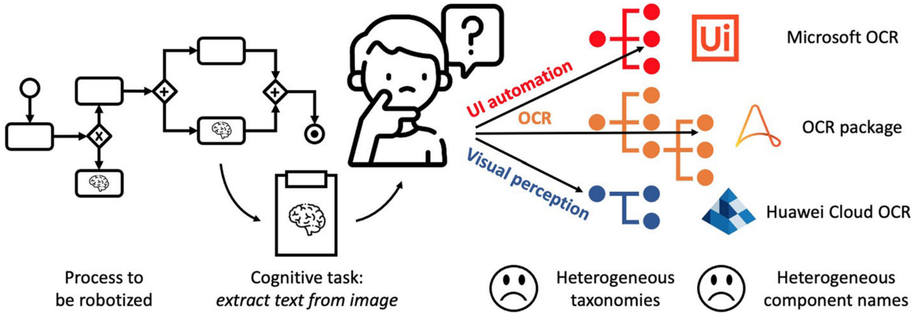


Fig. 1 Problem motivation

and classifications that is present in different platforms obstructs the selection of the most suitable components for solving a specific cognitive task and (2) it is tough to train team members in a particular discipline, such as language detection or text recognition in scanned documents, as there are no standard names or classifications of cognitive RPA components. Moreover, the ABBYY company, whose business line is based on digital intelligence for RPA, highlights that the lack of knowledge in developing RPA projects is one of the factors that lead to project failure [1].

To find the appropriate RPA component, methods like full-text search fail because the information found is typically incomplete since it does not include all the related components due to the heterogeneity in RPA component names and classifications. Moreover, they provide information on isolated components instead of components grouped and organized by similarities. Conversely, a taxonomy-based approach would allow to easily compare components with similar functionalities and, hence, facilitate the selection of the component that best suits the required need. This fact becomes even more relevant by considering that, due to the increasing interest in RPA, many beginning RPA developers do not have much experience in the field.

1.2 Contribution

In this work, a proposal for supporting the users in the development of a cognitive RPA project is presented. More precisely, an incremental method to automatically generate taxonomies from cognitive RPA platforms and experts is proposed. Such taxonomies can be dynamically adapted when necessary.

In previous work [21], the initial aspects of this research were presented. Furthermore, a method for systematically constructing a taxonomy of cognitive RPA components to support users in developing an RPA project was introduced. However, the current work greatly enhances such previous work by addressing a fundamental Primary Research Question (PRQ) not considered so far:

(PRQ) Does the proposed approach improve the support given to users during the development of a cognitive RPA project?

Intending to answer PRQ, this paper significantly extends and improves our previous work by:

- Extending the proposed method to improve the management of real-world use cases from industry,

- Developing a proof-of-concept tool, *CRPAsite*[6] from now on, that provides support to the proposed method,
- Validating *CRPAsite* by applying it to real-world use cases from industry, and
- Analyzing the related work by considering the foundations of some of the proposed methods for systematically reviewing the literature [17, 18, 24].

These extensions widely broaden the applicability of the proposed method. Moreover, they show that it can be successfully applied to real-world use cases in the industry.

The current work is framed within the AIRPA project,[7] which is focused on the integration between AI techniques and RPA. Such a project is the result of a collaboration between academics (ES3 research group[8]) and industry (Servinform S.A. company).

Finally, note that this work is closely related to some of the contemporary RPA challenges identified in [26] for future research, e.g., (1) *methodological support for implementation* and (2) *systematic design, development, and evolution* of RPA.

The rest of the paper is organized as follows. Section 2 describes a running example; Sect. 3 describes the proposed method for the systematic construction of a taxonomy of cognitive RPA components. Section 4 shows the result that is obtained after applying the proposed approach to the running example. Section 5 presents the empirical evaluation of this proposal. Section 6 includes a systematic review of literature on related topics while Sect. 7 discusses the limitations observed in the current work. Finally, Sect. 8 concludes the paper and describes future work.

2 Running example

In this paper, a set of use cases is selected from the industry, specifically from the execution of the AIRPA project in the context of business process outsourcing (BPO).

From these use cases, we select one of them from the banking sector as a running example. To be more precise, the selected use case is related to finding the degree of similarity between two text strings. The aim is to check whether two banking transactions have the "same" concept to verify whether an invoice has been paid. Figure 2 shows an example of this use case, in which it is necessary to compare the strings "2018/2–117" and "2018–2–117". For this purpose, a cognitive comparison component is used, which returns a 99% similarity result.

After reviewing the components of the main platforms that solve this problem (cf. Table 1), we could find the *fuzzy match* package in *Automation Anywhere*[9] (hereafter AA), which does not have any classification or category. In *BluePrism*,[10] we could find the *Approximate Matching* component, classified as *Natural Language Processing, Workflow, and Decision engines*. Finally, in *UiPath*,[11] there is no official component to solve this use case. It is necessary to browse third-party components in their marketplace,[12] which have a different classification

[6] Available at: https://github.com/RPA-US/crpasite.

[7] Available at: https://www.servinform.es/wp-content/uploads/2021/10/Servinform-AIRPA-Publicidad-web-corporativa.pdf.

[8] ES3 Group website: https://es3.us.es.

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

[10] Available at: https://digitalexchange.blueprism.com/dx/search.

[11] Available at: https://docs.uipath.com/activities.

[12] Available at: https://connect.uipath.com/marketplace/all-components.

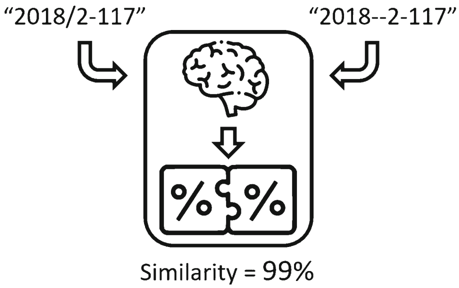


Fig. 2 Illustration representing the real use-case selected for the running example

Table 1 Components that are provided by the main RPA platforms related to the running example

RPA platform	Related components	Related categories
AA	Fuzzy match	Fuzzy match (package, not category)
BluePrism	Approximate matching	Natural language processing, workflow, and decision engines
UiPath	Novigo solutions—string similarity	String similarity (tag, not category)

from the official support ones. In such a marketplace, we find the *Novigo Solutions—String Similarity* component as a solution. For this, we consider *String similarity* as its category, although, in the UiPath marketplace, the components are not related to categories but tags. In summary, we could observe that similar components are related to solving the same cognitive task (i.e., comparing text strings) that are both named and classified differently.

3 Systematic construction of a taxonomy of cognitive RPA components

This section details the method proposed for systematically generating a taxonomy (i.e., *AI-RPA taxonomy*) from the cognitive components provided by different RPA platforms and experts. Section 3.1 details the structure of the resulting taxonomy. Section 3.2 defines the relations that are given between the categories according to their characteristics. Section 3.3 describes different roles that are involved in the process of creating a taxonomy. Section 3.4 gives insights into the process of creating as well as updating categories within a taxonomy.

3.1 Structure for the AI-RPA taxonomy

The proposed approach considers a simple tree structure for the resulting taxonomy. This structure comprises a root node (that has no meaning itself) and the rest of the nodes that correspond to taxonomic categories. For each category, the knowledge source is identified and stored in the AI-RPA taxonomy (cf. Definition 1). The knowledge sources considered are (1) RPA platforms and organizations and (2) human knowledge that experts in such fields provide. For instance, in the running example described in Sect. 2, the sources are both the UiPath, BluePrism, and AA platforms, as well as experts from Servinform S.A. and ES3 group. All the nodes of the tree can be updated by following the procedures detailed in Sect. 3.4.

Definition 1: An **AI-RPA taxonomy** *AI-RPAT = (KSs, TCs, CTs, CCs, IFs, OFs)* consists of:

- ***Knowledge Sources(KSs)***: a set of strings that corresponds to the names of the knowledge sources which are contained in the *AI-RPAT*.

Example 1 In the case of the running example, the knowledge sources correspond to the platforms consulted, which are UiPath, BluePrism, and AA, as well as to Servinform S.A. and ES3 group (hereafter SVF-ES3).

- ***Taxonomic Categories*** (***TCs***): a tree of strings whose nodes correspond to a taxonomic category. The following operations can be applied to *tc*, where *tc* corresponds to one of the taxonomic categories of *TCs*:
  - $parent(tc) \rightarrow tc'$, where tc' is the parent taxonomic category of *tc* in the tree.
  - $terms(tc) \rightarrow \{ct_1,..,ct_n\}$, where ct_i is an equivalent category term (see below) to *tc*. Thus, $terms(tc)$ keeps the synonymy relationships between its elements.
  - $repr(tc) \rightarrow ct$, where *ct* is the category term that corresponds to its representative term.
- ***Category Terms(CTs)***: a set of strings that represents the names of the categories. Different category terms can be associated with the same taxonomic category. However, only one of them will be selected as its representative term. The latter is used to establish the name of the corresponding taxonomic category. A category term comprises the following operations:
  - $knowlSource(ct) \rightarrow ks$, where *ks* corresponds to the knowledge source of *ct*.
  - $taxCat(ct) \rightarrow tc$, where *tc* corresponds to the taxonomic category it is associated with.
  - $isRepresentative(ct) \rightarrow b$, where *b* is a boolean that will be true if it is the representative term of its taxonomic category. In other words, it returns true if the category term corresponds to the one associated with a taxonomic category, i.e., it represents the rest of the category terms defined as equivalent terms.
  - $characteristics(ct) \rightarrow ls$, where *ls* is a list of strings with all descriptions of the characteristics (see below) associated with it.
  - $inputFS(ct) \rightarrow ls$, where *ls* is a list with the names of the input format supported associated with it.
  - $outputF(ct) \rightarrow ls$, where *ls* corresponds to a list with the names of the output format associated with it.
- ***Category Characteristics(CCs)***: a set of strings that corresponds to the descriptions of the category characteristics. Therefore, the components that belong to a specific category need to fulfill its related characteristics.
- ***Input Format Supported(IFs)***: a set of strings that corresponds to the names of the input types that can support the related categories.
- ***Output Format(OFs)***: a set of strings that corresponds to the names of the output types of the related categories.

The relations between the elements included in Definition 1 are depicted in the class diagram of Fig. 3. It is important to note that each taxonomic category has one or more related properties. For instance, for a single category term, there may be several *category characteristics* defining the characteristics that need to be fulfilled by a component for being considered part of such a category. Furthermore, each category may support more than one input format, and hence, it may have more than one *input format supported* associated.

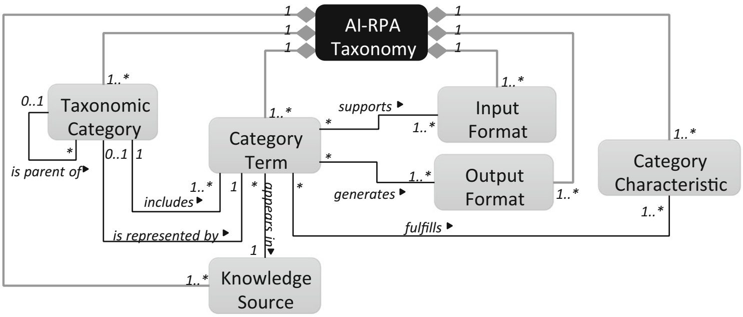


Fig. 3 Class diagram for AI-RPA taxonomy

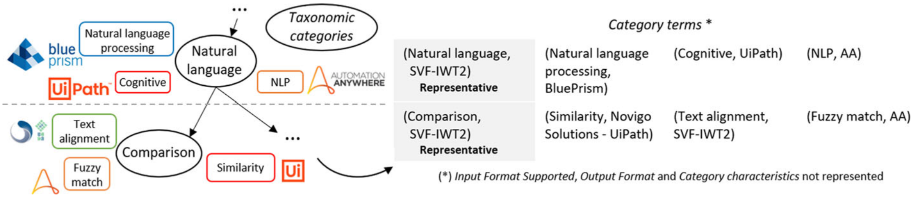

Natural language processing
blue prism
Cognitive
UiPath
Natural language
Taxonomic categories
NLP
AUTOMATION ANYWHERE
Text alignment
Comparison
Fuzzy match
Similarity
Ui
Category terms *
(Natural language, SVF-IWT2) Representative
(Natural language processing, BluePrism)
(Cognitive, UiPath)
(NLP, AA)
(Comparison, SVF-IWT2) Representative
(Similarity, Novigo Solutions - UiPath)
(Text alignment, SVF-IWT2)
(Fuzzy match, AA)
(*) Input Format Supported, Output Format and Category characteristics not represented

Fig. 4 Generation of a unified source of knowledge

Note that each taxonomic category contains one (and only one) representative term (cf. Fig. 4). The other category terms associated with the taxonomic category are considered non-representative terms because they are equivalent to the representative one, i.e., the taxonomic category.

Example 2 Following the use case presented in Sect. 2, SVF-ES3 identify a new category called *"Comparison"*. The components of that category are characterized by comparing text strings and returning the percentage of similarity or coincidence between these strings. Such category—together with all its related properties—is included in the taxonomy. In this case, the input format supported corresponds to *text*, while the output format corresponds to *a percentage*.

As characteristics, we could extract "*It takes as input two text strings to be compared*" and "*It returns as output a percentage of similarity*".

On the other hand, we consider *Similarity* as a term equivalent to *Comparison*, so this will be registered as a non-representative term, being a synonym of the representative category term *Comparison*.

This whole structure that forms the taxonomy revolves around classifying cognitive components. In this context, cognitive RPA components are considered software packages or web services that solve a specific activity within the operation of a robot, imitating human thought, action, or reasoning for its execution, using artificial intelligence techniques. These components are not part of the tree structure of the taxonomy but are attached to one or more categories. To be more precise, each category will have a series of attached components so that several categories may have the same components (i.e., the same component can be located in several categories in the taxonomy). In this way, the taxonomy supports component classification, and, in turn, new components serve to motivate the creation of new categories in the taxonomy.

In this context, an AI-RPA component is defined (cf. Definition 2).

Definition 2 An AI-RPA component consists of:

- ***Name***: the name of the component.
- ***List of characteristics***: both in terms of functionality and specification. At least one of these characteristics must match some of the characteristics of the taxonomic categories.
- ***List of input parameters***: including the format of each parameter and whether it is optional.
- ***Output format***: the format of the output(s) that the component returns, where some of the outputs may be returned depending on the input parameters.
- ***Main scope of business application***: the industrial sectors where the component's functionality has the most remarkable applicability. This attribute will be chosen from the following values: (1) *Financial and banking services*, (2) *Insurance Companies*, (3) *Telecommunications*, (4) *BPO*, (5) *Manufacturing*, (6) *Health*, (7) *Retail/e-commerce*, or (8) *Public sector*. In addition, it is possible to select more than one value. Finally, it is also possible to select the *Cross-industry* value proposed for a component considered transversal to all sectors.
- ***Satisfaction level***: the level of user satisfaction with regarding to the component, measured on a scale from 1 to 5. This way, every person that uses a component can store an assessment related to such component. This is stored as a recommendation.
- ***Count of selection***: number of times the component has been consulted. This metric allows finding the components with the highest degree of interest in the community.
- ***Provider***: the component supplier that corresponds to the RPA vendor (this role will be described later in this paper).
- ***Product***: can be either a code package or the credentials to make calls to a web service that returns the corresponding results. If the component corresponds to a web service, each API specialization will be translated into a new component instance.

Components and categories have an N: N relationship between them, that is, a component can be associated with several categories, and a category can be associated with several different components. Thus, relationships between component characteristics and category characteristics are established. Note that it provides the RPA developer with an additional information layer related to the components. Therefore, she will be able to filter components by category and different aspects, i.e., business area, providers, satisfaction level, input and output formats, or knowledge source, among others. In addition, it can be checked whether a component belongs to a category. For this, it is necessary to check if one of the component characteristics corresponds to, at least, one of the characteristics of the category.

3.2 Relationships between categories

Each category is defined through a set of characteristics. Due to the tree structure of the taxonomy, each category will correspond to one node. This way, all categories will have a parent category.[13] This relationship means that a specific category contains both its own characteristics and the characteristics of all its (direct and indirect) predecessors in the tree. Furthermore, two categories will be siblings—of the same level—if they have the characteristics of their ancestors in common.

[13] The categories of the first level would have a "null" parent category, so the root node has no associated attributes, (i.e., neither category terms nor characteristics).

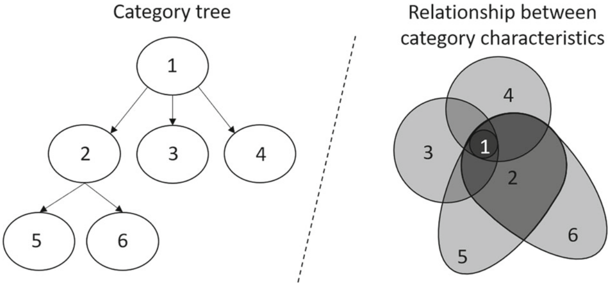


Fig. 5 Relationship between categories and characteristics

Example 3 Figure 5 shows a tree (in fact a subtree) in which the numbers correspond to the id of taxonomic categories. Categories 2, 3, and 4, that belong to the same level, have the same parent node (i.e., 1). Hence, 2, 3, and 4 have all the characteristics of 1. Moreover, they may include additional common characteristics. In the case of categories 5 and 6, they include all the characteristics of nodes 1 and 2. In the example depicted in Fig. 5, categories 5 and 6 only have in common the characteristics of their predecessors. Note that even categories of different levels (such as 3 and 5) can share some common characteristics.

For each component of the category, the characteristics are defined by selecting them from the list of taxonomy characteristics. A component belongs to a category if it matches at least one of its characteristics. In this way, to determine the appropriate category for a component, the taxonomy tree should be analyzed from top to bottom, comparing the characteristics of such component with the ones related to each node.

3.3 Roles involved in the generation of the taxonomy

Within the definition of an *AI-RPA taxonomy*, three essential roles are involved: (1) the *RPA developer*, that is the person that interacts with the taxonomy to the highest degree, (2) the *RPA vendor*, that uses the taxonomy to find the optimal categorization for its components, pursuing to capture their target audience by having well-classified components, and (3) the *moderator*, that is in charge of managing all the proposals and actions carried out over the taxonomy.

All taxonomy updates will be made through proposals. This means that any user interacting with the taxonomy will be able to create a proposal that a moderator will review. This way, the moderator is in charge of determining whether the category proposal is well-constructed to either accept or reject it (cf. Fig. 6). The moderator acts as an intermediary party for the acquisition of knowledge. When a related conflict occurs between two and more users, the moderator is in charge of collaboratively resolving such conflict to reach an agreement about the knowledge to be incorporated into the taxonomy. The proposed approach does not propose any concrete method to address this kind of conflict. However, many related methods could be easily applied in these situations (e.g., [9, 11]).

RPA developers and vendors will contribute their industry knowledge as proposals to the unified knowledge source that forms the taxonomy.

On the one hand, the interest of RPA vendors in the proposed approach is related to increasing the accessibility and visibility of their components. To achieve this, vendors must properly register their components. This is carried out as follows for each component to be

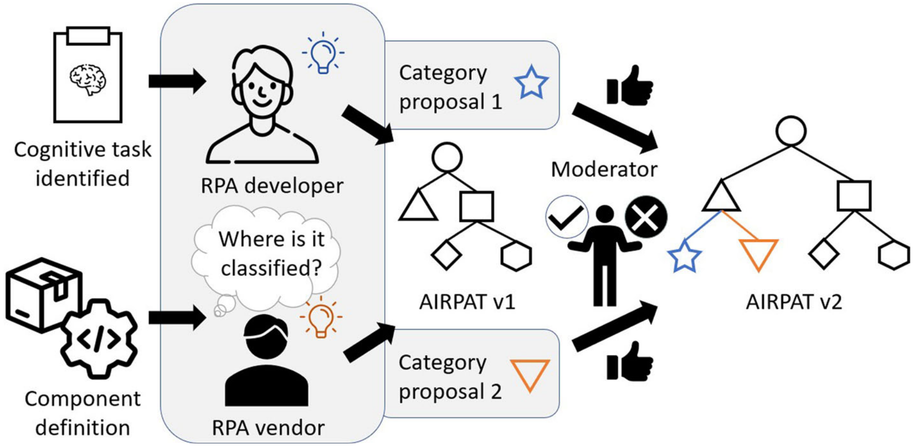


Fig. 6 How the roles are related to evolve the taxonomy? Motivation of RPA developers and vendors proposals, and dependence of moderator's approval

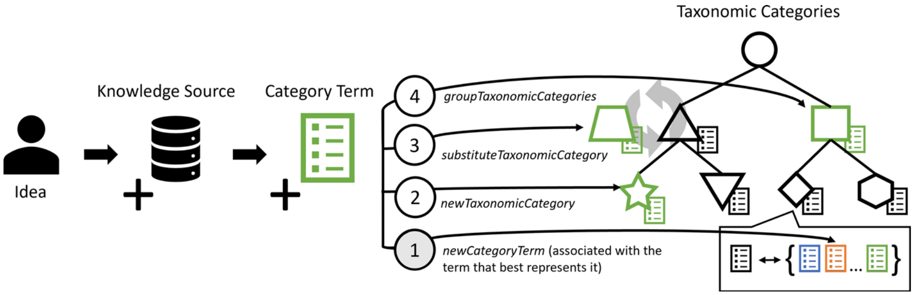

Taxonomic Categories
Idea
Knowledge Source
Category Term
4 groupTaxonomicCategories
3 substituteTaxonomicCategory
2 newTaxonomicCategory
1 newCategoryTerm (associated with the term that best represents it)

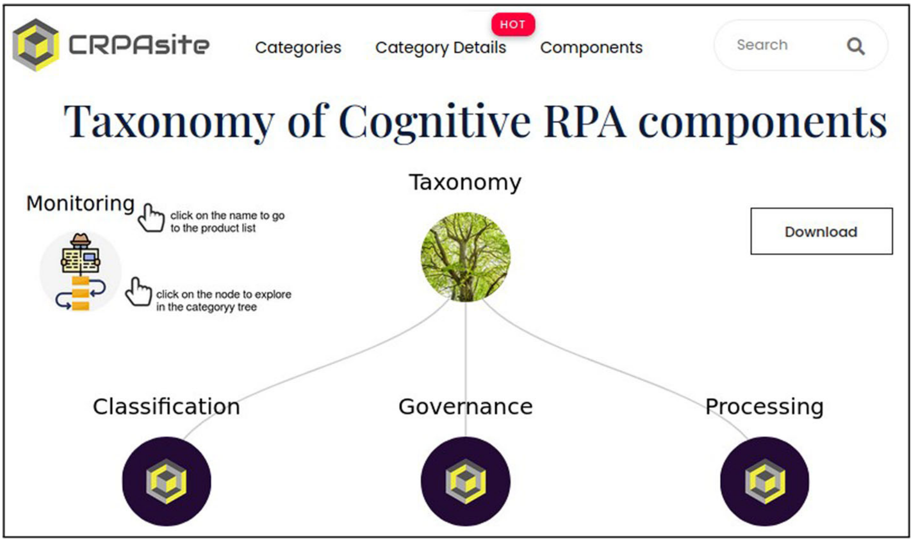
Fig. 7 Procedure to introduce new categories proposals in the taxonomy based on conditions described in Definition 3

included. First, the vendor looks for a category of the taxonomy that is suitable for such a component. If this category is found, the component is included within such category. Otherwise, the vendor may propose a new category for such a component to enhance the existing taxonomy.

On the other hand, the RPA developer can both take advantage of the existing taxonomy to find the component that solves her problem, regardless of the platform, and register new category proposals (cf. Fig. 7). Thus, each category proposal will produce a new version of the AI-RPA taxonomy. Therefore, the more use is given to the taxonomy, the more tools will be available since its approach is incremental.

To provide support to the users involved in this process, a proof-of-concept tool[14] (cf. Fig. 8) has been developed. This tool is based on the proposed approach.

[14] Available at: https://github.com/RPA-US/crpasite.



Taxonomy of Cognitive RPA components

click on the name to go to the product list

click on the node to explore in the categoryy tree

Fig. 8 Proof-of-concept tool based on the proposed approach

3.4 Managing categories in the taxonomy

An *AI-RPA taxonomy* can be initiated or evolved by following one of the two alternatives that are explained as follows. Firstly, when a new category is identified after analyzing a specific knowledge source (cf. Sect. 3.4.1), e.g., a category used in a company's internal operations or used by an RPA provider. Secondly, when there is a component that cannot be classified in the current state of the taxonomy, and, thus, it is necessary to define a new category in the taxonomy (cf. Sect. 3.4.2).

It should be noticed that, in the case that the taxonomy has not been yet initialized (i.e., the taxonomy is empty), the most natural way of initializing the taxonomy is to define the first level by including several *TCs*. It is advisable to include categories as much broad and general as possible in this first level. Otherwise, it leads to a slower process to evolve it.

Algorithm 3.1 How to evolve the taxonomy.

```
input CT $ct$, KS $ks$, TC $tc$, Action $a$, (optional) TCs $tcToBeMoved$
input/output AI-RPA Taxonomy $at$

if $ks \notin at.KSs$ then
    $at.KSs = at.KSs \cup \{ks\}$
end if
switch $a$ do
    case $newCategoryTerm$ do
        $at.CTs = at.CTs \cup \{ct\}$
        $terms(tc) = terms(tc) \cup \{ct\}$
    end case
    case $newTaxonomicCategory$ do
        $at.CTs = at.CTs \cup \{ct\}$
        $newTaxCateg = new\ TC(ct)$
        $parent(newTaxCateg) = tc$
        $at.TCs = at.TCs \cup \{newTaxCateg\}$
    end case
    case $substituteTaxonomicCategory$ do
        $repr(tc) = ct$
    end case
    case $moveTaxonomicCategories$ do
        for $tc'$ in $tcToBeMoved$ do
            $parent(tc') = tc$
        end for
    end case
end switch
```

3.4.1 New category motivated by a taxonomy term

A category will be considered equivalent to another one when the user who registers the proposal considers that the characteristics of both categories are analogous. In other words, the list of characteristics of the first category is included in the second one or vice versa. We will exclude the more technical or specific characteristics that do not restrict its functionality to make this comparison.

Whenever it is necessary to include or update any category, regardless of the level of the tree, the procedure followed is depicted in Alg. 3.1. This algorithm takes as input the following parameters: (1) the category term *ct* to be included in the taxonomy; (2) the knowledge source *ks* where it comes from; (3) the associated taxonomic category *tc*, whose semantic differs depending on the action to be taken (cf. Def. 3); (4) the Action *a* to be taken, whose values are detailed in Definition 3; and (5) the list of taxonomic categories *tcToBeMoved*, obtained after a manual search, that will be moved as children of the new one. The latter corresponds to an optional parameter that has a value only in the case that the *a* parameter takes the value *moveTaxonomicCategories*.

Definition 3 An ***Action*** in AI-RPAT is an enumeration of the possible modifications the taxonomy may suffer. An *Action* can correspond to:

1. ***newCategoryTerm***: this action serves to include a new category term *ct* in the taxonomy, i.e., a synonym of an existing taxonomic category that better represents the concept. In this case, the parameter *tc* corresponds to the taxonomic category that better represents *ct* (cf. line 5 of Alg. 3.1).
2. ***newTaxonomicCategory***: this action serves to include a new taxonomic category (i.e., TC) in the tree. It is added as a child of an existing taxonomic category, so it would be added as a leaf node. This is done when there is no equivalent category in the tree. In this case, the parameter *tc* corresponds to the most suitable parent of the taxonomic category that is going to be added (cf. line 9 of Alg. 3.1).
3. ***substituteTaxonomicCategory***: this action serves to replace the representative category term of an existing taxonomic category. This action is useful when the new term conveys better the concept of a taxonomy category. In this case, the parameter *ct* corresponds to the new representative term of the taxonomic category *tc* (cf. line 15 of Alg. 3.1).
4. ***moveTaxonomicCategories***: this action serves to move one or several taxonomic categories as children of another *tc* in the tree. In this case, the parameter *tc* corresponds to the tree node where the other taxonomic categories of *tcToBeMoved* will be moved to. Each of the *tcToBeMoved* is iterated to establish *tc* as its parent (cf. line 19–21 of Alg. 3.1). This action is useful to move a taxonomic category from one place to another in the tree, or to group several of them as children of a *new* one. For the latter, two actions will be necessary (i.e., create *newTaxonomicCategory* and *moveTaxonomicCategories*) as shown in the following example.

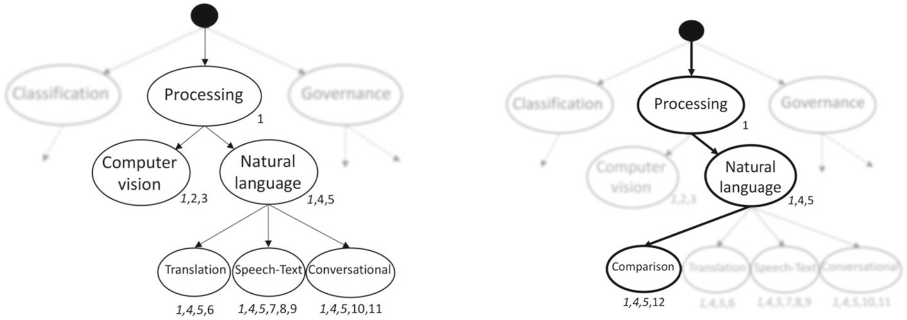
Example 4 Let *Translation*, *Speech-Text*, and *Conversational*, be three sibling taxonomic categories that are children of the *Processing* taxonomic category. If *Translation*, *Speech-Text*, and *Conversational* are desired to be grouped in a new common taxonomic category called *Natural language*, two actions need to be taken. First, the *newTaxonomicCategory* action is performed to include *Natural language* as a child of *Processing*. For this, $ct =$ “$NaturalLangu-age$” and $tc =$ “*Processing*”. And second, the *moveTaxonomicCategory* action is performed to move *Translation*, *Speech-Text*, and *Conversational* to become children of *Natural language*. For this, $tc =$ “*NaturalLanguage*” and $tcToBeMoved =$ [“*Translation*”,“$Speech - Text$”,“*Conversational*”]. The resulting relation between all the aforementioned nodes is shown in Fig. 9a.

Thus, once all the elements to construct the AI-RPA taxonomy have been defined, the incremental process has been described above in textual and graphic form (cf. Fig. 7).

3.4.2 New category motivated by a component

If the proposal of a new category in the taxonomy is motivated by a component, the procedure does not change, but it requires some previous steps to those described in Sect. 3.4.1.

Indeed, the category must be included in the same way in the tree. In the previous case (i.e., when the proposal is motivated by a taxonomy term), all the information related to the category is present. Conversely, when the proposal is motivated by a component, all the information related to the new category must be retrieved.



(a) Part of the taxonomy where the parent the new category is located.

(b) Taxonomy obtained after proposing of the *Comparison* category.

Fig. 9 Modifying the taxonomy by including a new taxonomic category. Numbers corresponding to *category characteristics* are shown in Table 3

That means that a component can be located in any taxonomy node, even if it is not a leaf node. This hierarchical structure will evolve dynamically, including new categories, for which the above steps are followed.[15]

If the tree is explored without finding any category that shares at least one characteristic with our component, it will be required to create a new category proposal to classify it. To do this, the component and its properties will be defined, extracting the essential attributes to define the category proposal. It is necessary to determine the characteristics of the component and to extract some of them for the category proposal. An example of the procedure is shown below.

Example 5 Suppose *Novigo Solutions*' String Similarity component[16] is identified. First, the definition of the component is created according to the structure described in Sect. 3.1:

- Name: String similarity.
- Characteristics: "Helps you to compare two strings quantitatively", "Includes the functionality of the Normalized Levenshtein, Jaro-Winkler, Metric Longest Common, Subsequence (MLCS), Cosine Similarity, Jaccard index, and SorensenDice-Coefficient algorithms", "Built using String Similarity.NET, a library implementing different string similarity and distance measures".
- Input parameters: First string, Second string, Algorithm.
- Output format: Number corresponding to the percentage of similarity.
- Main scope of business application: Cross-industry.
- The satisfaction level and the count selection will always be initialized to 0 and take values as RPA developers use it.
- Provider: Novigo solutions.
- Product: NovigoSolutions.StringSimilarity.1.0.0.numpkg

Then, the knowledge source is extracted. It corresponds with (1) the provider, **UiPath**, (2) the input format supported, which is a **string**, (3) the **identifier** for the type of algorithm, and

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

[16] Available at: https://connect.uipath.com/marketplace/components/novigo-solutions-string-similarity.

(4) the output format, which is a **number**. Later, the name of the category (i.e., **Similarity**) is selected. Depending on the provider's consideration, this name will correspond to the name of the component or a variation of it.

Regarding the characteristics of the category, it is desired to select those that define what the component does and not the ones that describe their specifications. In this example, in contrast to technical characteristics, such as the algorithm that the component can implement or the library with which it is built, "Helps you to compare two strings quantitatively" is selected.

4 Applying the proposed approach to the running example

A piece of the taxonomy that was defined in our previous work [21] is illustrated in Fig. 9a. The numbers associated with the nodes in the figure correspond to the identifiers of their characteristics, which are detailed in Table 3. This section describes how such taxonomy evolves due to the inclusion of a new category, as explained as follows. Such a new category is related to the Example 5.

The process to include the category motivated by the Novigo Solutions component—*Similarity*—in the taxonomy is detailed. First, the corresponding type of Action (cf. Definition 3) need to be checked. To do so, it is required to check if there is a taxonomy category that can be considered equivalent to *Similarity*. Considering that there are no taxonomic categories that represent its concept, the action *newTaxonomicCategory* is applied. That is, the similarity category is included as the child of the taxonomic characteristic that has more characteristics in common with it.

At the first level, some questions need to be answered (cf. Table 2). Reviewing the responses, only one branch is considered to be analyzed discarding the remaining. Then, it is necessary to check if the parent of *Similarity* could be located in such a branch.

From the proposal motivated by the component, it was identified as a characteristic "Helps you to compare two strings quantitatively". Based on this, as mentioned above, it was checked that, at the first level, *Similarity* shared the *Processing* characteristic "The output is obtained by transforming or modifying the input". Then, the characteristics of their children were studied. Here, *Computer Vision* lacks common characteristics with *Similarity*, since *Similarity* will never take images or documents as input and does not extract visual information. Moreover, the following characteristics of *Natural Language* were compared: "It takes as input text or audio" and "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". They are implicit and, therefore, shared between *Natural Language* and *Similarity*. Just

Table 2 Analyzing the most suitable branches when applying the proposed approach to the running example

Question	Response
Do the characteristics of classification fit with the characteristics of similarity?	No
Do the characteristics of governance fit with the characteristics of similarity?	No
Do the characteristics of processing fit with the characteristics of similarity?	Yes

Table 3 Category Characteristics of the resulting taxonomy of similarity

Id	Category characteristics	Description
1	Processing	The output is obtained by transforming or modifying the input
2	Computer vision	It takes as input an image or a document
3	Computer vision	It extracts from the input, concrete information that is visually inferred
4	Natural language	It takes as input a text or audio
5	Natural language	It transforms the input according to some of its. characteristics, such as language or format, or interprets the input to obtain a coherent output to it
6	Translation	As its name suggests, performs an interpretation of its input to translate it into a specified language
7	Speech-text	It extracts the message contained in the entry and transforms it into another format
8	Speech-text	If it is an audio input, it transforms the input into text
9	Speech-text	If it is a text input, it transforms it into audio
10	Conversational	It takes an input from interaction with a human through language, either written or spoken
11	Conversational	According to the input, it interprets its meaning and generates as output a coherent response
12	*Comparison*	*Helps you to compare two strings quantitatively*
13	*Comparison*	*It returns as output a percentage of similarity*

because *Similarity* extends these characteristics, it is necessary to include it as a new taxonomic category. Moreover, we will select *Natural language* as its parent, being instantiated as follows: Taxonomic Category: Similarity.

- Category Term: the same name “Similarity”, representative = true, knowledge source = *Novigo Solutions*.
- Category Characteristics: its characteristics and those of its ancestors, which corresponds to 1,4,5 (cf. Table 3) and “*Helps you to compare two strings quantitatively*”.
- Input Format Supported: String, Identifier.
- Output Format: Percentage.

Then, the moderator analyzes this category proposal to decide accepting or reject it. If it is accepted—the selected option for this example—, *Similarity* becomes part of the taxonomy.

Once the *Similarity* category was included, a new situation bursts into the state of the taxonomy. SVF-ES3 identified the *Comparison* category and wanted to include it in the taxonomy. To do so, the procedure described in Sect. 3 is followed. Therefore, the taxonomic categories are checked, looking for a term equivalent to the new category. Then, they are checked and *Similarity* is considered as meeting these conditions. Later, it was realized that *Comparison* better represents the concept than *Similarity*, since this category has analogous characteristics to those of *Comparison*. They only differ in that *Comparison* establishes an additional characteristic (i.e., “It returns as output a percentage of similarity”).

Such a characteristic establishes a higher level of specification on the category without changing or restricting its functionality. Therefore, it was decided to replace it with the category defined as follows:

- Taxonomic Category: Comparison
- Category Term: the same name, representative = true, knowledge source = SVF-ES3.
- Category Characteristics (cf. Table 3): 1,4,5,12,13.
- Input Format Supported: String, Number, Identifier.
- Output Format: Percentage.

To replace *Similarity* in the tree, it was included the category term associated with *Comparison* already defined, with *representative = True*, pointing to the taxonomic category of *Similarity*. It was replaced the name of the taxonomic category by *Comparison*, and in the category term that was associated with *Similarity*, it was updated to establish that *representative = False*. After this, *Similarity* will be stored as a term equivalent to comparison, and, assuming the approval of the proposal by the moderator, the taxonomy tree will be as depicted in Fig. 9b.

After including the new category in the taxonomy, it is possible to carry out the search process for the component in the category and check that our new category fits with the component characteristics.

As an example, to determine the category or categories of a component like the one defined in example 5, it is necessary to follow the next steps:

1. For each level, it is necessary to check if there is a category that contains at least one of the characteristics of the component. In this case, *Processing* is the only one that meets this condition.
2. It is necessary to proceed with the next level and to perform the same check, obtaining that *Natural Language* is the only one that has common characteristics with the aforementioned component.
3. It is necessary to repeat this process as many times as necessary (once for the considered example). Note that from *Translation*, *Comparison*, *Speechtext*, and *Conversational*, there is only one category (i.e., *Comparison*) that includes common characteristics with the considered component. Therefore, this component will be attached to the *Comparison* category.

5 Empirical evaluation

As explained in Sect. 1.1, this work is motivated by the fact that nowadays, the development of an RPA project is often a time-consuming, error-prone, and very tedious process. Therefore, as aforementioned, the following *PRQ* is addressed in this work: *Does the proposed approach improve the support given to users during the development of a cognitive RPA project?* With the goal of answering *PRQ*, three research questions (RQs) are investigated in the current empirical evaluation:

(RQ1): Does the proposed approach speed up the development of a cognitive RPA project?

(RQ2): Does the proposed approach reduce the number of errors that are given during the development of a cognitive RPA project?

(RQ3): Does the proposed approach improve the user experience during the development of a cognitive RPA project?

Herein, the case study protocol defined by Brereton et al. [6] is followed to improve its rigor and validation. In summary, the experiment is based on the resolution of a series of questionnaires by a group of professionals. In this survey, the leading RPA platforms and the current proposal are evaluated.

As proposed in [6], the rest of the section is organized as follows. Section 5.1 describes a real-world industry scenario where the study is carried out. In such a scenario, various use cases have been selected, as explained in Sect. 5.2. To perform the experiments, the questionnaires related to these use cases are sent to the selected subjects (cf. Sect. 5.3) that are involved in the experiments. Section 5.4 explains the design of the evaluation, while Sect. 5.5 details the execution. Thereafter, Sect. 5.6 includes the data collection. Finally, Sect. 5.7 analyzes the results that are obtained while Sect. 5.8 identifies some threats to the validity of the approach.

5.1 Case selection

This empirical evaluation is based on various industrial use cases derived from the AI-RPA project. The reasons why these use cases have been selected are the following:

- They require the integration of AI techniques within RPA, i.e., they contain cognitive tasks.
- They are based on real-world problems which are obtained from the industry.
- The selection of the RPA platform that provides the required component is not imposed or limited to a single one.

The use cases of the AI-RPA project are considered as suitable scenario since, beyond fulfilling the above criteria, they have been conducted by a highly experienced company, Servinform S.A. In addition, this company is involved in developing many RPA projects that include cognitive tasks, for which it considers components provided by different RPA platforms. More precisely, the AI-RPA project considers the following three RPA platforms that offer cognitive components: UiPath,[17] AA,[18] and BluePrism.[19] Therefore, such platforms have been selected for the experiments.

This first version of the taxonomy is the result of the collaboration of two RPA developers and two reviewers experts in the field of RPA.

During one month this team has studied the three platforms and generated the taxonomy, which has also been taken as the starting point for the development of *CRPAsite* platform. The result is a taxonomy[20] with 17 categories with 18 components, selected with the criterion that there must be at least one component per category. Some of these components can be classified in several categories, which are shown in Fig. 10. This figure shows the taxonomic categories in white circles and the number of components associated with each category in the black circles. In this *AIRPA taxonomy*, it is possible to classify components in nodes that are not leaves, e.g., *Natural language*, and it can be seen that the sum of the components is more than 18, since there are components associated with more than one category. Specifically, *AWS Textract* (Computer Vision & Detection—Elements), *Batch-OCR-Engine* (Computer Vision & Detection—Elements), and *IBM Watson Text Analysis* (Detection—Language & Natural Language).

[17] Available at: https://docs.uipath.com/activities.

[18] Available at:https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/using-the-workbench/cloud-commands-panel.html.

[19] Available at: https://digitalexchange.blueprism.com/dx/search.

[20] Available at: https://doi.org/10.5281/zenodo.5575218.

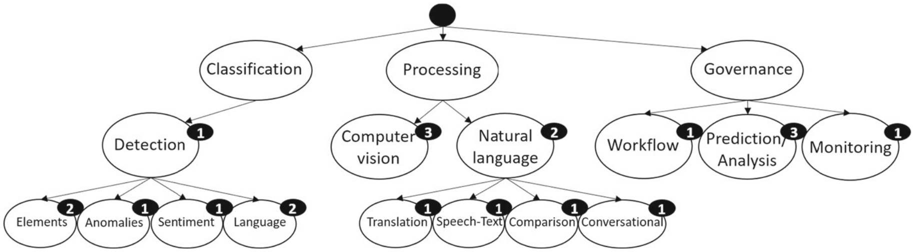


Fig. 10 Taxonomy that is used for the empirical evaluation

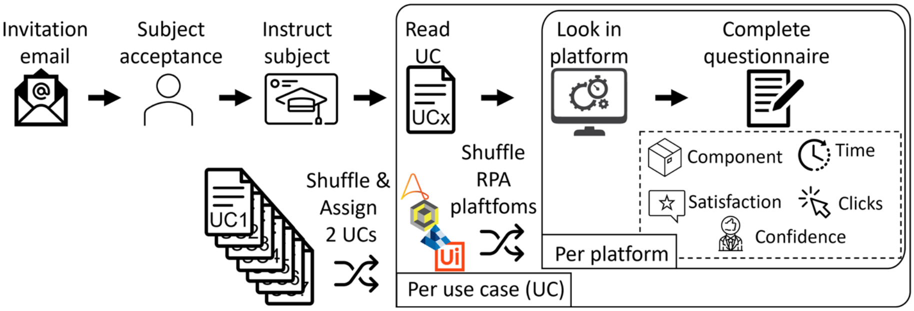


Fig. 11 Procedure followed by each subject during the empirical evaluation

For a month, some people have taken the three platforms and generated the taxonomy, which is also the result of a research project. The result is 17 categories with 18 components, some of which are classified in several categories and which are shown in Fig. 11.

5.2 Objects

The use cases selected for the experiments originate from a real-world project from industry, i.e., the AI-RPA project. From the set of use cases that are documented in such project, 7 were selected as *objects* for the evaluation (cf. Table 4). The use cases were selected by considering some important characteristics, such as diversity and representativeness.

5.3 Subjects

The subjects that are involved in the experiments are 12 IT professionals. Slightly more than half (i.e., 7 of them) are employees from the IT team of the Servinform company. The 5 remaining ones are employees that come from the academic sector. For the experiments, RPA experts are not required. Nonetheless, subjects should have at least a low or moderate knowledge of RPA and AI techniques. The profile of such employees is detailed in Table 5.

Table 4 Selected use cases from the AI-RPA project

Use case id	Company context	Description
(UC1)	Electric	Predicting the estimated reading of a data set from a customer's meter reading history
(UC2)	Banking	Detect which images taken to the meters contain personal images (e.g., faces) to avoid storing and sharing personal data without consent
(UC3)	Banking	Determine the degree of similarity of a set of characters (i.e., invoice identifier) within a text string (i.e., the concept of the bank movement) to verify whether an invoice is paid
(UC4)	Telephony	Detect anomalies in the development of the operators' working day during the processing of contracts. For instance, an unusually long duration is possibly caused by leaving the processing of the invoice open while the operator is away for a break
(UC5)	Retail	Extract information in a structured way from orders that come in PDF format without following any particular structure
(UC6)	Banking	Extract data from a telephone conversation (e.g., bank account number, identifier, or tax number) for subsequent verification by the operator
(UC7)	Electric	Identify the text language of an order to be later processed by the appropriate person

Table 5 Profile of the subjects that are involved in the experiments established in

Profile feature	Subjects' information
IT rol	29% (i.e., 2) of the subjects are *junior programmers*, 43% (i.e., 3) are *senior programmers* and 14% (i.e., 1), represent both *junior analysts* and *project leaders*
Academia rol	60% (i.e., 3) of the subjects are *junior researchers* and 40% (i.e., 2) are *senior researchers*
Educational level	50% (i.e., 6) of the subjects are *graduates*, 33% (i.e., 4) are *MSc*, and 17% (i.e., 2) are *PhD*. Moreover, 18% (i.e., 2) of the graduates are pursuing a master's degree
IT Sector experience time	8% (i.e., 1) of the subjects have *fewer than 1 year*, 58% (i.e., 7) have *more than 1 and fewer than 5 years* and 17% (i.e., 2) are both *more than 5 and fewer than 10 years* and *more than 10 years*
RPA experience time	60% (i.e., 7) of the subjects have *fewer than 1 year*, 40% (i.e., 5) have *more than 1 and fewer than 5 years*
AI techniques experience time	50% (i.e., 6) of the subjects have *fewer than 1 year*, 50% (i.e., 6) have *more than 1 and fewer than 5 years*

5.4 Design

The selected use cases are distributed among the subjects in such a way that each use case is managed by 3 different subjects. The subjects are selected by considering as much diversity as possible in the user profiles that solve the same use case. When managing a specific use case, each subject needs to answer questions related to the search process of the related cognitive component in different RPA platforms. To ensure that obtained results are not influenced by the order of appearance of the four different platforms in the questionnaire, such order is such a way that each platform appears the same number of times in the first, second, third, and fourth place.

With the goal of answering the aforementioned RQs, some evaluation metrics are proposed. First, the analysis and comparison of the values that are obtained for the following metrics are considered very helpful to answer whether the proposed approach speeds up the development of a cognitive RPA project, i.e., to answer (RQ1):

- *#clicks*: average number of mouse clicks that the subjects involved in the experiments need for finding the desired cognitive components when using CRPAsite as well as when using other existing platforms.
- *t*: average number of seconds that the subjects involved in the experiments takes to find the desired cognitive components when using CRPAsite as well as when using other existing platforms.

Second, the analysis and comparison of the values that are obtained for the following metrics are considered very helpful to answer whether the proposed approach reduces the number of errors that are given during the development of a cognitive RPA project, i.e., to answer (RQ2):

- *%ErrCat*: average number of errors that are given related to selecting the wrong category for the considered task when using CRPAsite as well as when using other existing platforms.
- *%ErrComp*: average number of errors that are given related to selecting the wrong component for the considered task when using CRPAsite as well as when using other existing platforms.

Third, the analysis and comparison of the values that are obtained for the following metrics are considered very helpful to answer whether the proposed approach improves the user experience during the development of a cognitive RPA project, i.e., to answer (RQ3):

- *Sat*: average degree of user satisfaction when using CRPAsite as well as when using other existing platforms. This metric can take a value from 1 to 5.
- *Conf*: average degree of user confidence when using CRPAsite as well as when using other existing platforms. This metric can take a value from 1 to 5.

The platforms that are considered for all the aforementioned evaluation metrics are BluePrism, UiPath, Automation Anywhere, and CRPAsite. According to this, Table 6 includes the response variables that are defined for the current empirical evaluation together with the related research questions.

Table 6 Research questions and related response variables of the empirical evaluation

Research question (RQ)	Response variables (RV)
(RQ1)	$\#clicks_{BP}$, $\#clicks_{UiP}$, $\#clicks_{AA}$ and $\#clicks_{CRPA}$ *. t_{BP},t_{UiP},t_{AA} and t_{CRPA} *.
(RQ2)	$\%ErrCat_{BP}$, $\%ErrCat_{UiP}$, $\%ErrCat_{AA}$ and $\%ErrCat_{CRPA}$ *. $\%ErrComp_{BP}$, $\%ErrComp_{UiP}$,$\%ErrComp_{AA}$ and $\%ErrComp_{CRPA}$ *.
(RQ3)	Sat_{BP}, Sat_{UiP}, Sat_{AA} and Sat_{CRPA} *. $Conf_{BP}$, $Conf_{UiP}$, $Conf_{AA}$ and $Conf_{CRPA}$ *.

*BluePrism (BP), UiPath (UiP), Automation Anywhere (AA), and CRPAsite (CRPA)

Table 7 Data that is obtained for the response variables

RVs	RQ1		RQ2		RQ3	
Platform	#clicks	t(s)	%ErrCat	%ErrComp	Sat	Conf
UiPath	16.85	286.54	77.00	77.00	2.15	2.85
AA	18.56	280.50	75.00	81.00	2.06	2.50
BluePrism	12.77	150.69	62.00	62.00	1.85	3.54
CRPAsite	8.38	90.43	24.00	24.00	3.71	3.71

5.5 Execution

The experiments were conducted in October 2020. The way to proceed with the experiments is described graphically in Fig. 11 and explained as follows. First, the subjects receive an *invitation email* that formally asks them to participate in the current research. This email contains the links to the online forms associated with two different use cases that must be completed. Then, in the case that the *subject accepts*, by clicking on each link, she is welcomed and *instructed* with the description and instructions of the experiment.

The experiment consists of filling out a questionnaire for each of the four RPA platforms (i.e., AA, CRPAsite, BluePrism, UiPath), for each of the two use cases. These two use cases are randomly assigned by shuffling the seven existing ones. In the same way, RPA platforms questionnaires are assigned and shuffled to prevent them from always appearing in the same order.

The questionnaire consists of a series of questions aimed at collecting the metrics defined in this experiment, which are number of *clicks*, *time* spent, degree of subject *satisfaction* and *confidence* when using the corresponding platform, as well as the *component* that has been selected. The error percentage per category and per component will be calculated depending on the component that has been selected. These questionnaires are shown to the subjects as part of the online form, which should be completed and submitted as the last step. The answers of the subjects are properly recorded to be later analyzed.

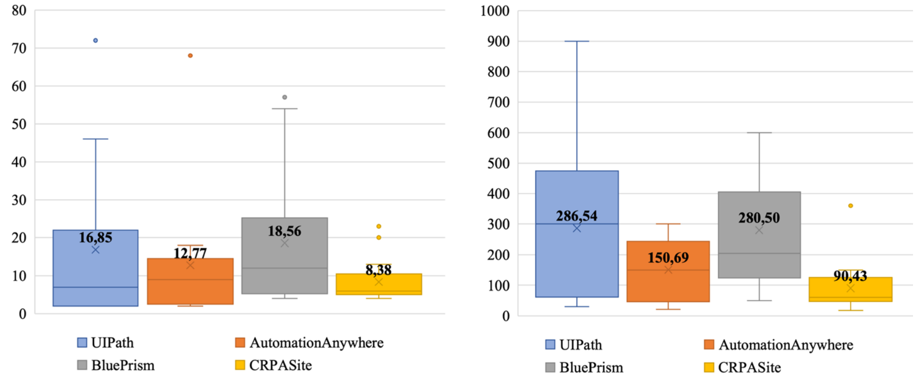
5.6 Data collection

This section collects the results that are obtained after performing the experiments. These results are shown in Table 7. As can be observed, such table includes the average values recorded from the response variables, for all considered use cases, and related to (RQ1), (RQ2) and (RQ3), respectively.

5.7 Data analysis

The analysis of the collected data is illustrated in Figs. 12a, b, 13a, b, 14a and b. As can be observed, two kinds of values are represented in these figures:

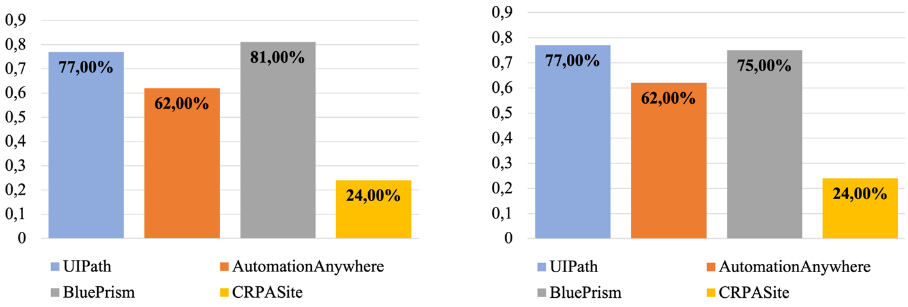
- For the response variables related to (RQ1) and (RQ3), the median values that are obtained are depicted in a box and whisker chart (cf. Figs. 12a, b, 14a, and b).
- For the response variables related to (RQ2), the average values that are obtained are depicted in a bar chart (cf. Figs. 13a and b).



(a) Average number of mouse clicks (i.e., *#clicks*).

(b) Average number of seconds (i.e., *t*).

Fig. 12 Box and whisker responding to RQ1



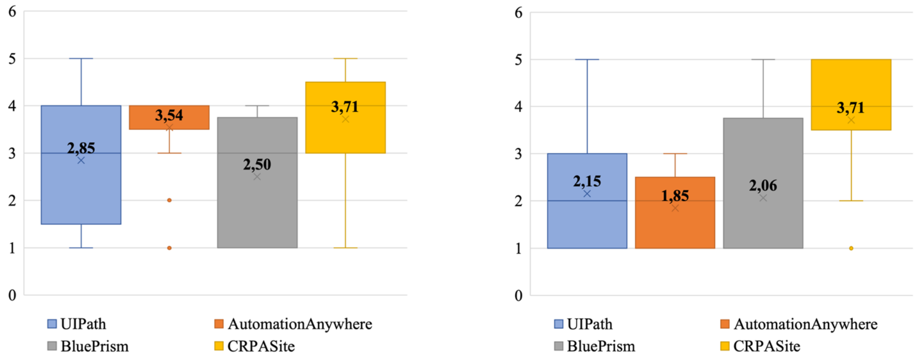
(a) Average number of category errors (i.e., *%ErrCat*).

(b) Average number of component errors (i.e., *%ErrComp*).

Fig. 13 Box and whisker responding to RQ2

Regarding (RQ1), as shown in Fig. 12a, although the minimum number of *#clicks* is not obtained in the *CRPAsite* platform, it gets the lowest value for the median. Moreover, the lowest maximum is registered by the *CRPAsite* platform. Furthermore, the number of outliers (i.e., the atypical values) that can be observed in the *CRPAsite* platform is meager compared to the values that are obtained in the remaining platforms. In addition, when analyzing the time *t* invested by the subjects in finding the suitable category and component for each use case, it is possible to observe that the *CRPAsite* platform reveals as the best option (cf. Fig. 12b). Note that it represents the lowest minimum and maximum values, even having a relatively high outlier. Considering all these findings, (RQ1) can be answered as true.

Concerning (RQ2), Fig. 13a and b shows the percentage of errors that are made by the subjects when selecting categories and components for a specific use case. As can be observed, the values obtained for the response variables related to these two aspects (i.e., *%ErrCat* and *%ErrComp*) are pretty similar.



(a) Average degree of user satisfaction (i.e., *Sat*).

(b) Average degree of user confidence (i.e., *Conf*).

Fig. 14 Box and whisker responding to RQ3

In fact, only one of the subjects failed to select the suitable component, having selected the correct category. Moreover, the *CRPAsite* platform reduces the value of *%ErrCat* and *%ErrComp* by half compared to the second classified platform (i.e., *Automation Anywhere*). Hence, this platform can be considered the best option, and (RQ2) can be answered as true.

The user experience related to (RQ3), measured by *Sat* and *Conf* variables, is represented in Fig. 14a and b. Concerning *Sat* (i.e., the satisfaction degree), *CRPAsite* can be again considered the best option since it gets both the highest median value as well as the highest maximum value. Regarding *Conf* (i.e., the confidence degree), *CRPAsite* can also be considered the best option since it has the highest minimum and maximum values. Moreover, the value of the median is almost double that of the value obtained by the second classification. Therefore, (RQ3) can be answered as true.

In summary, the main findings that can be highlighted after analyzing the collected data are:

- *CRPAsite* gets the lowest median values for #*clicks* and *t* response variables compared to the remaining platforms.
- *CRPAsite* gets the highest values for the median and the maximum related to *Sat* and *Conf* response variables compared to the remaining platforms.
- *CRPAsite* platform gets the lowest minimum values for *t*, *%ErrCat* and *%ErrComp* response variables compared to the remaining platforms.
- Even if there are outliers (as can be observed in #*clicks*, *t*, *Sat* and *Conf*), *CRPAsite* platform gets better results than the remaining platforms.

Considering that (RQ1), (RQ2), and (RQ3) can be answered as true, the primary research question (PRQ) can also be answered as true.

5.8 Threats to validity

As aforementioned, 12 subjects participated in the experiments. We are aware that this is a small sample. However, this sample size is not unusual for this kind of experiment due to the substantial effort to be invested per subject [7, 23]. Note that the selected subjects represent

Table 8 Keywords for performing the review of related work

Keyword	Plural and synonyms
Taxonomy	Taxonomies
RPA	Robotic process automation, process automation
AI	Artificial intelligence

an appropriate sample [2] since all of them have at least a low or moderate knowledge of RPA and AI techniques but differ in other aspects, such as IT role and educational level.

Moreover, we consider 7 use cases for the experiments, which is a relatively low number. In addition, all selected use cases are related to the BPO sector. Therefore, this may hamper result generalization. However, note that these use cases were selected considering some important characteristics, such as diversity and representativeness. What is more, the experiment is conducted over a taxonomy that includes 18 components distributed in 17 categories (cf. Fig. 10). That implies that most of the categories include a single component, which may have an effect on the results. Nonetheless, despite the relatively low number of components per category, we observed that the subjects expended most of the time surfing through the tree of categories instead of their component lists. Therefore, we expect an enclosed impact in the result when more components are considered.

Furthermore, the experiments focus exclusively on cognitive components.

Therefore, additional experiments are required to determine whether the same findings would apply to RPA components from other domains.

Lastly, the experiments are focused on the analysis of 3 RPA platforms. We are aware that considering a higher number of platforms would be helpful to generalize the study findings further.

6 Related work

There are various methods for systematically reviewing the literature, e.g., Systematic Literature Review (SLR) [17], Systematic Mapping Study (SMS) [24] or Tertiary Studies (TS) [18], among others. In this work, the foundations of these methods have been taken as a basis for reviewing the work related to this proposal. To be more precise, considering the PRQ defined in Sect. 1.2, this section aims to highlight which is the state-of-the-art concerning the use of dynamic taxonomies in the context of RPA and AI cognitive components.

To construct the queries to be executed in the digital libraries, some keywords were defined (cf. Table 8). In addition to these keywords, other synonyms (e.g., classification, ontology or terminology for the term taxonomy), were defined. However, these synonyms were finally discarded because they introduced too much noise in the results obtained. A query was constructed using the keywords of Table 8, and it was executed in Scopus[21] digital library. Because the results obtained were not very close to the present proposal, it was decided to divide the query into two contexts: RPA and AI. In addition, the keyword taxonomy was kept as a common point (cf. Table 9). For the RPA-related query, the appearance of the keywords was limited to title, abstract, and keywords. For the AI-related query, the appearance of the keywords was limited just to the title since abstract and keywords introduced too much noise in the results obtained. Both queries were limited to "Computer Science" area.

[21] Available at: https://www.scopus.com/search/form.uri?display=basic.

Table 9 Queries for performing the review of related work

Id	Query
1	TITLE-ABS-KEY ("Taxonomy" OR "Taxonomies") AND TITLE-ABS-KEY ("RPA" OR "Robotic Process Automation" OR "Process Automation") AND (LIMIT-TO (SUBJAREA, "COMP"))
2	TITLE("Taxonomy" OR "Taxonomies") AND TITLE ("AI" OR "Artificial Intelligence") AND (LIMIT-TO (SUBJAREA, "COMP"))

The queries of Table 9 were executed, obtaining 17 documents. The title and abstract of these documents were analyzed to check the alignment with the current approach. After this process, 12 documents were removed, remaining 5 primary studies to be in-depth analyzed. Finally, one more primary study that was identified in previous work [21] that did not appear in the results was added. Twelve characteristics were established to classify the primary studies:

- Characteristic: it indicates whether the approach defines characteristics in the taxonomy.
- Groups: it indicates whether the approach defines groups to organize the taxonomy.
- Source of Information: it states where the information to create the taxonomy comes from. Three options have been observed: (1) from a literature review, (2) from a single input (e.g., a given reference), and (3) from subject-matter experts.
- Creation Method: it defines how the approach creates the taxonomy. Two options have been observed: (1) from the own authors and (2) following a systematic method.
- Extension Mechanisms: it indicates whether the approach provides mechanisms to be extended.
- Collaborative Creation: it indicates whether the approach considers the collaboration of people when creating the taxonomy.
- Collaborative Extension: it indicates whether the approach considers the collaboration of people when extending the taxonomy.
- Type of Characteristics: it states what is used to classify the elements in the taxonomy.
- Type of Elements: it states what content the taxonomy is aimed for.
- Organization. It states how the elements in the taxonomy are distributed. Three options have been observed: (1) a hierarchical tree-like structure, (2) a 2-level structure with elements and groups, and (3) no structure at all, i.e., all the elements are in a single group.
- Applied: it states whether the approach includes the application of the taxonomy to any context.
- Validated: it states whether the approach includes any validation of the taxonomy.

As can be observed in Tables 10 and 11, the related work found fits very superficially with the proposal presented in this paper. More precisely, although all of them define characteristics or features in the proposed taxonomy, only half define groups or categories to organize it based on such characteristics. Moreover, none of the approaches includes the experts' opinions when creating the taxonomy. In turn, they rely on a single reference or in a broader literature review. Regardless of this source of information, none of the primary studies present a taxonomy that can be created systematically. They do not propose a way to extend it either, but the taxonomies presented are static. Consequently, these proposals are not collaborative. Nonetheless, we found a couple of works that partially cover some of these characteristics. On the one hand, [4] bases upon a set of principles—that could be extended—to derive the taxonomy, but the mechanism to do so is not provided. On the other hand, [15] creates its

Table 10 Classification framework for the related work (Part 1)

Work	Characteristic	Groups	Source of information	Creation methods	Extention Mechs	Collab. creation	Coallb. Extension
Beerbaum [4]	✓	✓	Literature	Own	∂	X	X
Arrieta et al. [3]	✓	✓	Literature	Own	X	X	X
Ding et al. [12]	✓	✓	Single input	Own	X	X	X
D'Rosario et al. [8]	✓	X	Single input	Own	X	X	X
IEEE Corporate [15]	✓	X	Single input	Own	X	∂	X
Davis et al. [10]	✓	X	Literature	Own	X	X	X
Current proposal	✓	✓	Literature + experts	Systematic	✓	✓	✓

✓ = true, X = false, ∂ = partial

Table 11 Classification framework for the related work (Part 2)

Work	Types of characteristic	Types of elements	Organaization	Applied	Validated
Beerbaum [4]	Risks	AI algs	2 levels	X	X
Arrieta et al. [3]	Internal feats	AI algs	Tree	✓	X
Ding et al. [12]	Observable feats	AI algs	2 levels	✓	X
D'Rosario et al. [8]	Issues	Cognitive RPA platforms	X	X	X
IEEE Corporate [15]	RPA concepts	Definitions	X	X	X
Davis et al. [10]	Observable feats	AI algs	X	X	X
Current proposal	Observable feats	Cognitive RPA components	Tree	✓	✓

✓ = true, X = false, ∂ = partial

taxonomy involving a working group and a committee, but this collaboration mechanism was not disclosed.

Regarding the type of the characteristics (cf. Table 11), two of the taxonomies deal with a concrete topic (i.e., risks and issues in RPA, respectively), while the others are more general. Nonetheless, although two of them use observable features of the elements to organize the taxonomy, none deals with cognitive RPA components but with general AI algorithms. It is important to note that, similar to the current proposal, [3] is the only proposal that provides a tree-like organization of the taxonomy. The other proposals organize their elements in a flat distribution of groups or do not provide groups at all. Furthermore, although two of the proposals describe the application of the taxonomy to a sample field, none of them presents any kind of validation.

To summarize, some close or partially close works related to the topic of this paper have been found in the literature. However, any of them propose an incremental taxonomy for the cognitive components in the RPA context.

7 Discussion and limitations

As previously mentioned, this work enhances our previous work by extending the proposed method to improve the management of real-world use cases, developing a related proof-of-concept tool, validating the developed tool, and performing a systematic review of literature on related topics. The limitations observed in each of these extensions are explained as follows.

Regarding the proposed method (and hence, the related proof-of-concept tool), note that the users establish the similarity degree among the characteristics of categories and the components. Hence, it may lead to unexpected human errors. In addition, the process of including a component within a specific category is subject to the fact that the user selects the characteristic that best defines such component from a set of characteristics provided by the tool. It may happen that the user does not find any characteristic that fits well with this component. In this case, she would need to propose a new related category. This may result in a high increase in the size of the taxonomy. However, this is palliated by the taxonomy's hierarchical nature generated through the proposed approach. Moreover, it is important to mention that all the decisions that imply changes in the taxonomy need to be approved or rejected by the moderators, which affects the collaborative nature of the taxonomy.

Regarding the systematic literature review on related topics, note that we did not perform a strict SLR itself. However, the foundations of SLR as a basis for reviewing the work related to this proposal have been considered.

Regarding the validation of the proposed approach, as detailed in Sect. 5.8, some limitations may hamper result generalization. First, the number of subjects involved in the experiments is not high. Nevertheless, this is not unusual for this type of experiment [7, 23], and we consider that the selected subjects are appropriate [2] by taking their profiles into account. Moreover, we consider a relatively low number of use cases related to the same sector. However, these cases were selected by considering diversity and representativeness, among others. In addition, the experiments focus exclusively on cognitive components and are performed over a small number of platforms. Hence, considering additional domains and platforms would be desirable to generalize the study findings.

Note that the proposed approach could also be used for different purposes. As an example, the lack of homogeneity in names and classifications of components is also a problem in the scenario related to collaborative distribution platforms. To be more precise, when an independent RPA vendor offers components on a collaborative distribution platform, such as UiPath Connect[22] or BluePrism Digital Exchange,[23] she must understand the classification that is followed by each platform to be able to place components within the correct category. Additionally, other difficulties may appear depending on the platform. For instance, in the

[22] Available at: https://connect.uipath.com/marketplace/all-components.

[23] Available at: https://digitalexchange.blueprism.com/.

case of UiPath, the components that the vendor provides are classified into groups that are not available for the categorization of UiPath Connect components.

8 Conclusions and future work

This work proposes an incremental method to automatically generate taxonomies from cognitive RPA platforms. This method aims to support both RPA developers in the phase of analysis and implementation of the robot and RPA vendors to place their components on a collaborative distribution platform. Thus, RPA vendors will be able to reach their target audience, which are RPA developers looking for a specific taxonomy category. Such taxonomies can be dynamically adapted when necessary. In previous work [21], the initial aspects of this research were presented. However, the current work greatly enhances such previous work by: (1) extending the proposed method to improve the management of real-world use cases from industry, (2) validating the proposed method by applying it to real-world use cases from industry, (3) developing a proof-of-concept tool (i.e., *CRPASite*) that is based on the proposed approach, and (4) systematically reviewing the literature. The proposed approach speeds up the development of a cognitive RPA project. Moreover, it reduces the number of errors and improves the user experience during this development. Therefore, it substantially improves the support given to users during such development.

Unlike previous related work (i.e., [3, 4, 8, 10, 12, 15]), the proposed approach does not propose a specific taxonomy but a method for systematically generating such taxonomy from the information that is provided by 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, in contrast with the previous related work, the proposed approach is focused on specific RPA platforms, i.e., on platforms that provide AI-based solutions. Moreover, a new aspect that is added compared to previous related work is the validation of the proposed approach. This validation demonstrates that *CRPASite* platform can be successfully applied to real-world industry use cases.

As for future work, some extensions are intended to be explored:

- Further generalizing the study findings by performing more experiments involving a higher number of subjects, a higher number of use cases, and additional RPA platforms.
- Extending the proposed approach to allow its application to RPA components from other domains.
- Developing a recommendation system for building and using the taxonomy for: (1) providing recommendations about how to introduce the elements (e.g., taxonomic categories, category terms, etc.) into the taxonomy, and (2) providing recommendations about the most suitable component for performing a specific task. For developing such a recommendation system, it would be required to express the characteristics of the categories as rules.

In conclusion, currently, there is no method or standard to support the search or supply of cognitive RPA components, and this method seeks to provide a solution to a problem that can be developed much further, as we see in these lines of future work.

Acknowledgements This publication is part of the project PID2019-105455GB-C31, funded by MCIN/AEI/10. 13039/ 50110001 1033/ and by the "European Union"; the project P20_00644, funded by Junta de Andalucía; and the project AIRPA (EXP001180 29/IDI-20190524-P011-19/E09) funded by the Centro para el Desarrollo Tecnológico Industrial (CDTI) of Spain. Moreover, the first author of this work is currently supported by the FPU scholarship program, granted by the Spanish Ministry of Education and Vocational Training (FPU20/05984).

Funding Funding for open access publishing: Universidad de Sevilla/CBUA.

References

1. ABBYY. State of Process Mining and Robotic Process Automation 2020. www.abbyy.com/en-us/solutions/process-intelligence/research-report-2020. Last accessed Feb 2023
2. Anslem S, Corbin J (1998) Basics of qualitative research: techniques and procedures for developing grounded theory. SAGE Publications, Thousand Oaks
3. Barredo Arrieta A et al (2020) Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 58:82–115
4. Beerbaum D (2020) Artificial intelligence ethics taxonomy—robotic process automation (RPA) as business case. In: European Scientific Journal
5. Biscotti F (2018) Market share analysis: robotic process automation, worldwide. https://www.gartner.com/document/3923903. Last accessed Feb 2023
6. Brereton P, et al (2008) Using a protocol template for case study planning. In: 12th International conference on evaluation and assessment in software engineering (EASE), vol 12, pp 1–8
7. Costain G (2008) Cognitive support during object-oriented software development: the case of UML diagrams. Ph.d thesis. ResearchSpace@ Auckland
8. D'Rosario M, D'Rosario C (2020) Beyond RoboDebt: the future of robotic process automation. Int J Strateg Decis Sci IJSDS 11(2):1–24
9. Dalkey NC, Helmer-Hirschberg O (1962) An experimental application of the delphi method to the use of experts. RAND Corporation, Santa Monica
10. Davis J, Hoffert J, Vanlandingham E (2016) A taxonomy of artificial intelligence approaches for adaptive distributed real-time embedded systems. In: 2016 IEEE international conference on electro information technology (EIT). IEEE, pp 0233–0238
11. Delbecq AL, Van de Ven AH, Gustafson DH (1975) Group techniques for program planning: a guide to nominal group and delphi processes. Management applications series. Scott, Foresman
12. Ding R-X et al (2020) Large-scale decision-making: characterization, taxonomy, challenges and future directions from an Artificial Intelligence and applications perspective. In: Information fusion, pp 84–102
13. Enríquez JG, et al. (2020) A unified model representation of machine learning knowledge. In: Journal of web engineering, pp 319–340
14. Fung HP (2014) Criteria, use cases and effects of information technology process automation (ITPA). In: Advances in robotics and automation
15. IEEE Corporate Advisory Group (2017) IEEE guide for terms and concepts in intelligent process automation
16. Jiménez-Ramírez A, et al (2019) A method to improve the early stages of the robotic process automation lifecycle. In: International conference on advanced information systems engineering. Springer, pp 446–461
17. Kitchenham B, et al (2009) Systematic literature reviews in software engineering—a systematic literature review". In: Information and software technology, vol 51.1 , pp 7–15
18. Kitchenham B et al (2010) Systematic literature reviews in software engineering—a tertiary study. Inf Softw Technol 52(8):792–805
19. Lacity M, Willcocks L, Craig A (2016) Robotizing global financial shared services at royal DSM. In: The outsourcing unit working research paper series
20. Le Clair C, et al (2019) The forrester wave™: robotic process automation. In: The forrester wave

21. Martínez-Rojas A, Barba I, González Enríquez J (2020) Towards a taxonomy of cognitive RPA components. In: Business process management: blockchain and robotic process automation forum, pp 161–175
22. NASSCOM. Cognitive RPA, The Future of Automation. https://www.nasscom.in/knowledge-center/publications/cognitive-rpa-futureautomation. Last accessed: Feb 2023
23. Nielsen J (1994) Estimating the number of subjects needed for a thinking aloud test. Int J Human Comput Stud 41(3):385–397
24. Petersen K, et al (2008) Systematic mapping studies in software engineering. In: 12th International conference on evaluation and assessment in software engineering (EASE), vol 12, pp 1–10
25. Slaby JR (2012) Robotic automation emerges as a threat to traditional low-cost outsourcing. In: HfS Research Ltd 1.1, pp 3–3
26. Syed R et al (2020) Robotic process automation: contemporary themes and challenges. Comput Ind 115:103–162
27. Tarquini T (2018) Practical robotics in insurance–the future is here already. In: The insurtech book: the insurance technology handbook for investors, entrepreneurs and fintech visionaries, pp 231–235
28. Willcocks L, Lacity M, Craig A (2017) Robotic process automation: strategic transformation lever for global business services?". J Inf Technol Teach Cases 7(1):17–28

Antonio Martínez-Rojas received his M.S. in software engineering at the University of Seville, Spain, and is currently working toward a Ph.D. in computer science at the same institution. He has demonstrated his expertise in the field with multiple high-impact publications in international journals and conferences. His research interests span across robotic process automation, business process management, decision model discovery, and task mining. He is an active participant in developing research projects funded by public funds, and he also collaborates closely with companies to transfer his research findings to practical applications.

Irene Barba holds M.S. and Ph.D. degrees in computer science at the University of Seville, Spain. She is an associate professor at the University of Seville. Her research interests include the optimized management of business processes, user support in flexible process-aware information systems, robotic process automation, constraint programming, planning, and scheduling. She has co-authored numerous publications in international journals and conferences. She has participated in several R&D projects funded by European Commission, National or Regional entities, and technology transfer projects with private companies. She has been member of the organizing and scientific committees of several conferences.

Carmelo Del Valle received his M.S. and Ph.D. degrees in industrial engineering from the University of Seville, Spain. He is an Associate Professor at the Department of Computer Languages and Systems of the University of Seville. His research interests include constraint programming, planning, scheduling, combinatorial optimization, business process management, and product lifecycle management. He has co-authored numerous publications in international journals and conference proceedings. He has participated in several R&D projects funded by European Commission, National or Regional entities, and technology transfer projects with private companies, acting as principal researcher in many of them. He has been member of the organizing and scientific committees of various international symposiums.

Andrés Jiménez-Ramírez is a lecturer and researcher at the University of Seville. He started his career as a software engineer in the R&D department of a Spanish consultancy firm. In 2010, he moved to the University, where he is a member of the Engineering and Science for Software Systems (ES3) research group. He is currently an associate professor at the Computer Languages and Systems department. His main research interests are process automation, flexible process modeling, and constraint programming. In these areas, he has co-authored more than 30 contributions to prestigious international conferences and journals and participated in 7 research projects and over 20 technology transfer projects involving IT companies.

José González-Enríquez José González Enríquez holds Ph.D. in computer science at the University of Seville since 2017. He is an associate professor at the Computer Languages and Systems department of the University of Seville. His research interests include robotic process automation, business process management, model-driven engineering, and testing. He participates in developing research projects financed by public funds and actively collaborates with companies to transfer his research, often acting as principal investigator. He has been part of the organizing committee of different international conferences and collaborated with many universities in countries such as Southampton, Cuba, and Berkeley. In 2015, he received the Innovation Award for his innovative activity in the field of research and development of the Platform for DynamicData Integration Andalusian Historical Heritage by Fujitsu Laboratories of Europe.