On Evolvability Issues of Robotic Process Automation (RPA)

Geert Haerens

Department of Management Information Systems
Faculty of Business and Economics
University of Antwerp, Belgium
Engie IT — Digital & IT Consulting
Email: geert.haerens@engie.be

Herwig Mannaert

Department of Management Information Systems
Faculty of Business and Economics
University of Antwerp, Belgium
Email: herwig.mannaert@uantwerp.be

***Abstract*—Robotic Process Automation (RPA) receives a lot of publication attention in business and academic publications. RPA has also become big business, as it offers a cheap, fast and non-intrusive solution for businesses who want to improve their process performance while not having to re-engineer their processes and/or overhaul their IT landscape. Current literature points out some limitations for RPA but does not go much further than some rules of thumb. Using the Normalized Systems (NS) theory – a theory to study modular structures' behavior under change – we can surface that RPA has serious evolvability issues. These evolvability issues have been observed as well by RPA practitioners. This paper contributes to both the value of NS to study evolvability and to point out the evolvability limitations of RPA, which are currently underrepresented in related research.**

Keywords*–*RPA; Normalized Systems; Evolvability

I. INTRODUCTION

RPA is popular in the landscape of business process optimization methods. While executing a business process, multiple applications may be used. In an ideal and fully digital world, each process step can be handled by an IT system. The IT systems can exchange information and trigger each other, without the need for human intervention during to execution of the business process. Most companies have not reached this level of digitalization. They may lack applications to handle a part of the process or have existing applications that are not suited for application interaction and triggering. In those cases, a human will bridge the gap between applications to keep the process going. The idea behind RPA is to replace a human, who is performing tedious and repetitive tasks within or between applications, by a software robot - also known as a bot. Like the human, the bot - a software program operating on the user interface of a computer system [1] - only acts via the user interface of the application(s) to manipulate information stored in one or more applications. Manipulations happen via mouse-clicks and keyboard-strokes, just like humans would. RPA is a kind of outsourcing of repetitive tasks to the computer. The bots are our new "Co-workers" [2]. They are part of the future digital workforce [3]. A bot does not replace the human. A bot takes the "robot" out of the human [4], meaning that repetitive/robotic tasks are no longer done by the human, allowing human resources to focus on more value-added activities.

RPA is realized by a client-based piece of software and includes both a design- and a runtime aspect. The design-time part of RPA is typically a low code environment that allows to define what the bot needs to do. The run bot is running on a client-based machine, performing activities according to the scenarios outlined during design time.

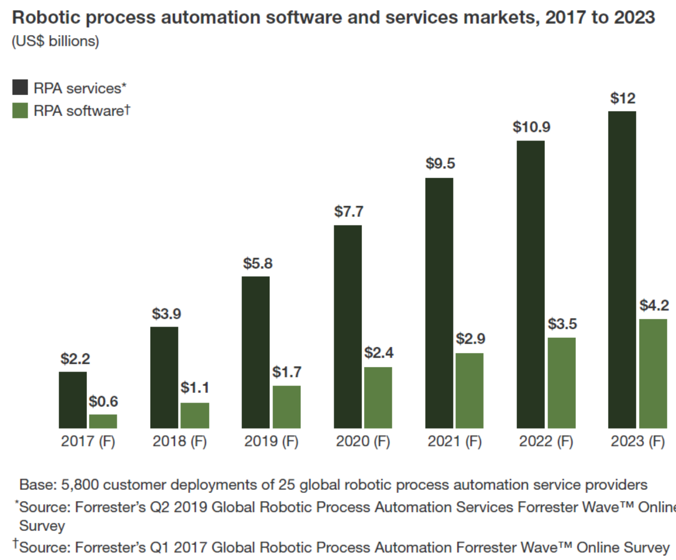

Figure 1. RPA market predictions according to Forrester - from [5]

RPA works well on cases where structured data is available as input and a clear, stable and standardized set of action rules exist, such that the outcome of the performed actions is unambiguous [4]. Typical targets for RPA are shared-service-center activities, which are highly standardized, such as financial, procurement, and HR Backoffice processes.

Process automation is widely accepted as a first step to the digital transformation of a company. Techniques, such as Business Process Management and Automation (BPM/A), have been around for some time. While those focus on fundamentally changing and continuously improving the process, RPA keeps the existing process and application landscape intact. RPA takes the valuable human resources out of the loop and replaces it by a bot. RPA promises faster, cheaper, and better execution of specific processes in a non-intrusive way. The RPA business is booming, both in terms of tooling which allow the design and run of bots, as in the services (consultancy) related to RPA, that include the selection of the right process for RPA, creation of the business case, setup of centers of excellence and operational maintenance of bots.

 ISBN: 978-1-61208-783-2

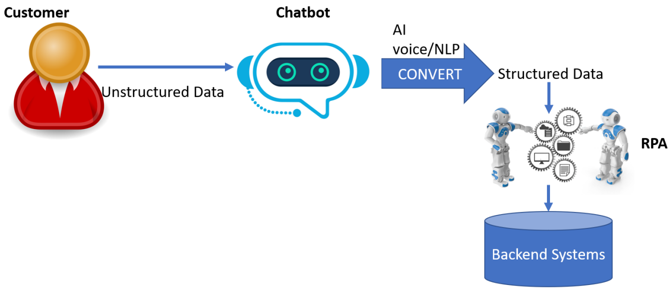

Figure 2. Combining AI (chatbots) and RPA for Customer Communications - from [8]

Figure 1 shows the predictions Forrester [5] is making with regards to the RPA market, both in tooling and services.

A great future is being projected for RPA, and that future becomes even more ambitious if RPA would be combined with Artificial Intelligence (AI) [6]. RPA can only work with structured data. AI holds the promise to convert unstructured data into structured data and continuously learn and improve in this. By combining both AI and RPA, processes previously out of reach for RPA become feasible candidates for automation. Figure 2 outlines a process that includes both AI at the front and RPA at the backend to interact with the customer [7].

Bright as this future may look, choosing the right process to apply RPA to, is the critical success factor. Existing literature limits itself to providing some rules of thumb. A critical aspect of choosing the right process for RPA, is the stability of the process. This aspect raises the question of how well RPA performs under a set of anticipated changes. The Normalized Systems theory [9]-[10] studies the effect of anticipated changes on modular systems. The theory uses concepts of classic engineering, such as system stability – Bounded Input = Bounded Output - and statistical entropy - possible microstates for a given macrostate - to determine the necessary conditions a modular system must adhere to, in order for the system to be stable under a set of anticipated changes. In this paper, NS will be used to study the impact of change on RPA [11].

The remainder of this paper is structured as follows. In Section II, RPA is being elaborated, including an overview of existing and related work. The Section also includes a basic introduction to NS. In Section III, the evolvability of RPA will be investigated using a simple but realistic process related to expense notes. In Section IV, two companies testify with regards to their RPA initiative. In Section V, the theoretical findings of Section III and the practical feedback from Section IV will be discussed. Finally, Section VI concludes and provides suggestions for further research.

II. Literature Study and Related Work

This paper focusses on the ability of RPA to cope with change and uses NS [11] theory as an analysis instrument. The next paragraphs will provide an overview of known RPA issues and a short introduction to NS.

A. Literature on RPA issues

In [12] it is stated that multiple publication can be found about the various benefits of RPA, based on real-life implementations - 68% of publications - but less on academic research

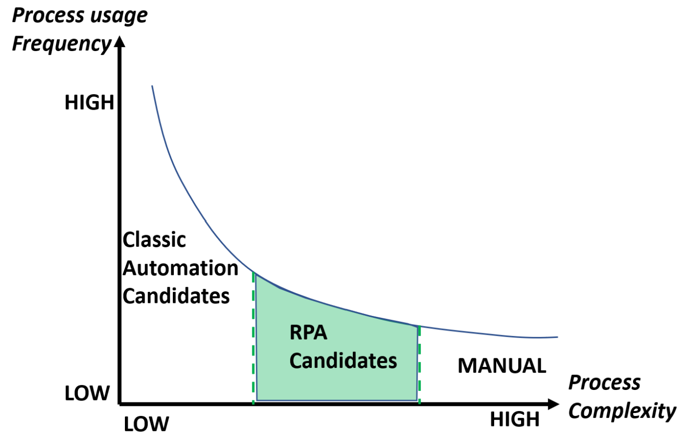

Figure 3. RPA processe candidates by process frequency and complexity - from [4]

on the topic -15% of publications). The remaining 18% are literature studies. The literature study [12] focused on publications that could provide insight to the following research questions: "RQ1: What is the current state and progress of RPA?", "RQ2: How is RPA defined and how does it relate to BPMS", and "RQ3: How is RPA used in practice according to the scientific literature".

RPA is taking up a good part of the current process management industry and is frequently a subject of analysis reports by Forrester and Gartner. Forrester analyst Craig Le Clain has made multiple reports on RPA, including observed limitations. The "Rule of five" [8] is an RPA design criteria that comes back a few times in his reports. The rule states that an RPA solution should limit itself of max five decisions, access to 5 applications, and should not contain more than 500 clicks. The main motivation for this rule of thumb, the limited rule capabilities, the static nature of the code, and vulnerability to application changes are given. According to Le Clain[6] [7][8], AI can help to overcome some of those issues. AI will reduce robot maintenance (auto-adjust to application changes), externalize decisions from the bot scripts, use unstructured data as input, and team-up with chatbots for data input. Some scenario's for auto-correction include AI as well. Changes to the application images/screens are to be detected by AI, and will adjust the bot script automatically. AI can be fed with information from outside of the RPA digital world, to understand and interpret the context of changes and sending alerts to bot control for issue that cannot be corrected with a high degree of certainty.

In [4], Jovanović et all point out the benefit of the non-intrusive nature of RPA. Business Process Management and Automation can only work if the applications used in a process have some integration points, like APIs, which allows the manipulation of data elements and execution of tasks. Those kind of integrations are more complicated and require higher programming skills compared to the low-code environment RPA often provides. With RPA, no adjustments of the existing applications are required, which is a compelling fact for businesses to choose RPA over BPM/A. Jovanovic et all [4] sum up properties of processes suitable for RPA:

- Low cognitive requirements

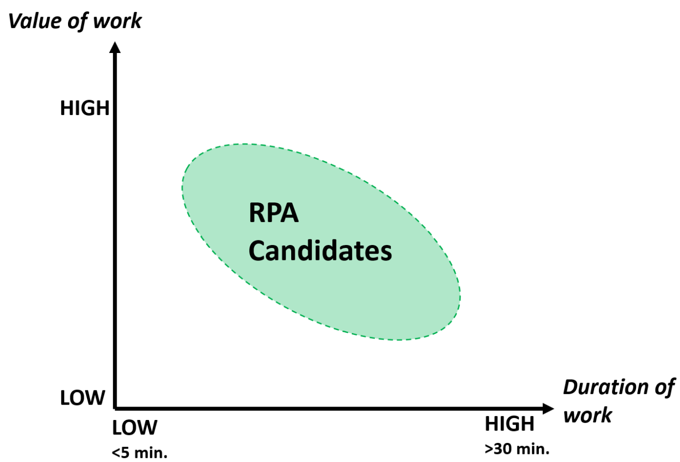

Figure 4. RPA processe candidates by value of work and duration of work - from [4]

- Access to multiple systems not required
- High Volume
- High probability of human error
- Limited exception handling

One may notice those are stricter compared to the “Rule of 5” of Forrester. Previous studies [2] show that supporting processing, such as those handled in shared service centers, are better candidates for RPA than the core (key) business processes. Supporting processes are more standardized and fall into the RPA candidates regions outlined in Figure 3 and 4. The paper of Jovanovic et all [4] concludes with a quote from Bill Gates [13]: ‘The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the Inefficiency’.

In [2], Osmundsen et all refers to work of Bygstad [14], which discusses the position of RPA in the IT organization. RPA can be seen as a personal productivity tool for business people, allowing them to automate parts of a process without having to go via the IT department. Such an approach is related to the setup of lightweight IT or Bimodal IT, as Gartner calls it. Business is able to self automate without having to startup big, lengthy, expensive, and sometimes frustrating IT projects. Classic IT departments (heavyweight IT) are of course, not happy with lightweight IT and often have a deep aversion against bots. They see bots as a poor man’s integration tool, not even worthy of the name IT solution. Bygstad [14] argues RPA should be part of lightweight IT. The business knows best its processes and will thus be more successful in configuring the bot. Lightweight IT should be loosely coupled to heavyweight IT. This does introduce additional challenges, such as the lack of control mechanisms around RPA, leading to spaghetti-solutions and automating the wrong processes. The lack of end-to-end process views leading to local optimization, not necessarily global optimization, is listed as a second challenge. To overcome those challenges, one can see RPA in lightweight IT as a way to foster innovation and build enthusiasm for digitalization, while tightening the relationship with heavyweight IT at a later point in time.

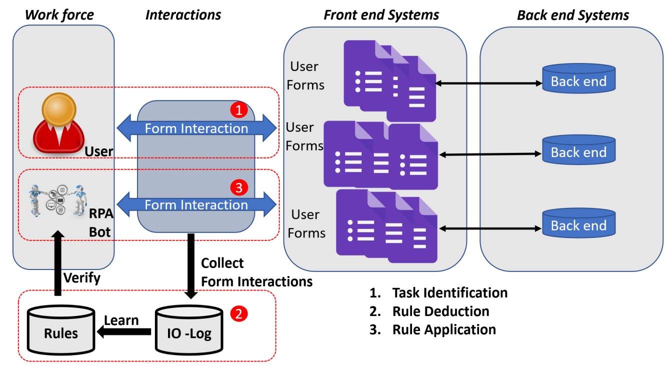

Figure 5. Method followed to deduce rule base, based on learing from human application interaction - from [15]

Goa et all [15] focus on the configuration of the bot. Instead of consciously creating the rule base the bot must use, they propose to learn the rule base from a human. An artifact is being proposed (see Figure 5) , which will deduce the rule base based on the interaction the human has with the application. After a while, the artifact will have sufficiently learned about the usage of the system to take over from the human. In their current approach, nothing is mentioned about re-training the bot in case of application changes and thus the impact of change on the bot.

In [16] the risks associated with RPA are divided into three categories: governance risks, technical risks, and process risks. The governance risks are related to the operating model associated with RPA: centralized, federated, or decentralized. Each operating model has different characteristics, which influence RPA maturity. The technical risks are about the impact of IT availability on RPA. If the technical side is not working, the process is directly impacted. The process risks are about the selection of the correct process to apply RPA to and the development steps to come to a working solution. RPA can execute repetitive work faster and with higher quality. However, if the process is not properly reflected or if it gets erroneous data as input, it will make mistakes more swiftly and with certainty [1]. Hence the need to select only processes that are well known and have stable and reliable input data.

To best of the authors efforts, no papers were been found that explicitly address the evolvability issues of RPA (main contribution of this paper). The literature that discusses the technical risks are the closets match. They describe the effects of change, not the root cause of these effects.

B. Introduction to NS

NS originates from the field of software development [9] [17] [10] . There is a widespread belief in the software engineering community that using software modules decrease complexity and increases evolvability. It is also well known that one should strive towards “low coupling and high cohesion”. The problem is that the community does not seem to agree on how exactly “low coupling and high cohesion” needs to be achieved and what the size of a module should be, to achieve low complexity and high evolvability. NS takes the concept of system theoretic stability from the domain of classic engineering to determine the necessary conditions a modular structure of a system must adhere to in order for the system to exhibit stability under change. Stability is defined as Bounded Input equals Bounded Output (BIBO). Transferring this concept to software design, one can consider bounded input as a certain amount of functional changes to the software and the bounded output as the number of effective software changes. If the amount of effective software changes is not only proportional to the amount of functional changes but also the size of the existing software system, then NS states that the system exhibits a Combinatorial Effect (CE) and is considered unstable under change. NS proves that, in order to eliminate CE, the software system must have a certain modular structure, where each module respects four design theorems. Those rules are:

- Separation of Concern (SoC): A module should only address one concern or change driver
- Separation of State (SoS): A module should have a state which is observable by other modules.
- Action Version Transparency (AVT): A module, performing an action should be changeable without impacting modules calling this action.
- Data Version Transparency (DVT): A module performing a certain action on a data structure, should be able to continue doing this action, even is the data structures has undergone change (add/remove attributes)

Only by respecting those rules, the system can infinitly grow and still be able to incorporate new requirements. While the four theorems mentioned above are used during design time, NS has additional theorems usable for run time as well. Making use of the concept of statistical entropy, NS derives the necessary condition for a system to be diagnosable, being the ability to determine the actual microstate of a system, given a certain macrostate. Formulated differently, the software is not working (macrostate) because module x is not working (a microstate). The necessary condition for this is summarized in the following theorem:

- Instance Traceability: The ability to know the state of an instance of a module at run time.

Although NS originates in software design, the applicability of the NS principles in other disciplines, such as process design, organizational design, accounting, document management, and physical artifacts. The theory can be used to study evolvability in any system, which can be seen as a modular system and drive design criteria for the evolvability of the system.

The environment in which RPA is applied can be split into three layers: process, application, and infrastructure. Each of those layers can incure change. Making use of the NS theorems, it can be determined whether those changes will have an effect on RPA proportional to the change, or to the change AND the system itself (a combination of process, application, application, and RPA). In the former case, the introduction of RPA should be declared stable under change; in the latter case, RPA should be declared unstable under change.

III. Investigating Evolvability of RPA

In the next paragraph, a simple process step in an expense note process will be presented, followed by the introduction of RPA to automate this process step. This section is continued with a paragraph on the general impact of change in the process environment and concludes with the specific impact of those changes to RPA, to evaluate the degree of stability of RPA with respect to anticipated changes.

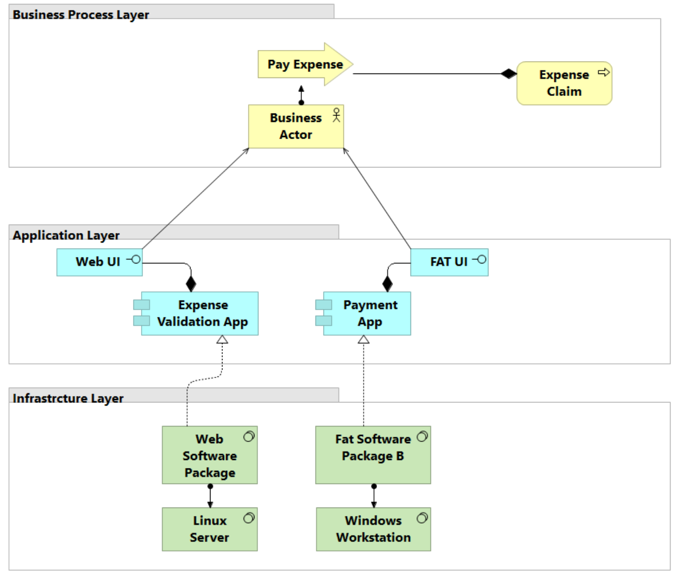

Figure 6. Refund process without RPA

A. Description of the process

Consider an expense note process. The process consists of an employee declaring his expenses, approval by the manager, and finally, reimbursement of the expense. Assume that a company has no IT system available, that has the possibility to detect the approval of an expense note by the manager, and a system that automatically performs the refund (money transfer). The company has an “Expense Validation” application and a “Payment” Application. A human (business actor) will connect to the “Expense Validation” application to see which expenses are currently flagged as “validated” and then use the “Payment” application to perform the actual money transfer to the expense claimer. The “Expense Validation” application is realized by a web application installed on a Linux host. The “Payment” application consists of a fat client application realized by a software package running on a Windows workstation. The UIs (Web and fat client) of both applications are used by the human business actor to perform the “Pay Expense” process step. A visual representation of the layered architecture, using ArchiMate [18], can be found in Figure 6.

B. Introducing RPA

The described process step would be a good candidate for RPA as the process step is simple, and all information required for deciding and launching the “Pay Expense” process step, is available in the “Expense Validation” and “Payment” application. The human Business Actor is being replaced by a bot, which will use the UI of both applications to perform the process step. The bot itself is a “bot Player” application, which is realized by RPA system software, which is being installed on a Windows workstation. A visual representation

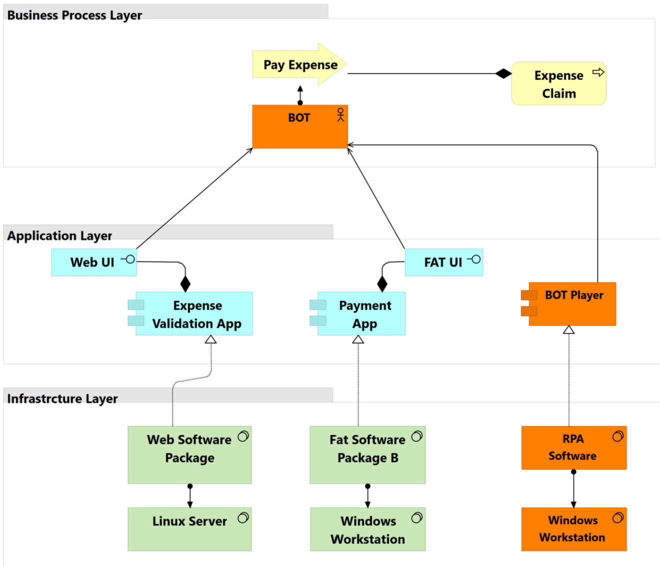

Figure 7. Refund process with RPA

of the layered architecture, including the bot, can be found in Figure 7.

C. Changes in the process environment

A lot of businesses struggle with change in general. Putting cultural change aside, making changes to a company often has unforeseen side effects, coined as ripple effects in NS, due to hidden couplings in the organization. Those couplings can be found in and between the organizational structure of the business, the processes inside the organization, the applications supporting the processes and the infrastructure supporting the applications. NS has studied the effects of change at the business process layer [19], the application layer [9] [17] [10], and the infrastructure layer [20]. In each of those layers, CE are present. They can be eliminated or mitigated by applying the NS principles by careful and conscious design of processes, applications, and infrastructure.

RPA is part of the process environment. It is part of an environment in which changes can ripple in all directions and with varying intensity. It should come as no surprise that a system as RPA, which works at the edge of the environment via the UI, will be impacted by changes to the process, application, and infrastructure layer.

D. Changes in the RPA environment

The environment in which the bot is working can be subject to the following changes:

- At the Business Process Layer, changes to the process can happen, such as the addition of an extra process step and the introduction of a new business actor. In the example process, an extra validation step could be added for expenses higher than a certain amount.
- At the Application Layer, changes to the application can happen due to changes in the process or changes to the software (the addition/removal of data objects, data object attributes, new process steps, new UI components). In the example process, expense claims higher than a certain amount needs to be explicitly selected and a particular transaction with this selection needs to be launched
- At the Infrastructure Layer, changes to hosts and system software can happen due to the scaling of the application, new system software releases, the usage of different compute resources. In the example process, changes to the OS version may lead to a higher screen resolution, resulting in a repositioning of UI elements on the screen.

The above changes can be anticipated over the life cycle of the Business Process. The changes will ultimately become visible to the Business Actor via the UI:

- New clicks and stokes to be performed in the UI of the application due to changes in the process.
- New clicks and stokes to be performed in the UI of the application due to addition, removal or relocation of information and action items in the UI.
- New location and/or size of action and information items due to new UI elements, UI look and feel and UI behavior.

The UI of the application is literally the only window on the process in the digital world. Human actors can act according to information available in the real world and the digital world. For instance, a change in the process could be explained via a communication letter, and the human actor would be able to understand and act on the corresponding application changes due to information provided outside of the digital world. A bot cannot do this and will require reprogramming to cope with the changing scenario. As the UI is the only window on the process in the digital world, the UI will reflect the aggregation of all change drivers possible in the environment. This is a clear violation of the Separation of Concerns principle of NS. Behind one UI element, multiple concerns may be hidden. Changes to the UI element can be due to several reasons. Without additional information outside of the digital world, the reason for the change and the appropriate action to take, cannot be determined. The fact that the UI is the aggregation of all change drivers also makes the diagnosability of RPA an issue. A change visible in the UI cannot be traced back to its origin (process, application, infrastructure) by only looking at the UI. The full-stack needs to be investigated. This is a violation of the Instance Traceability principle, leading to a CE.

From the above, the conclusion can be drawn that the environment in which RPA operates, being an environment in which the only interaction point with the process is through application UI's, is inherently unstable under change, as it violates both the Separation of Concerns and Instance Traceability design principles of NS. Although the impact of change to the RPA solution itself has not been studied, one can expect evolvability issues there as well. A new version of the bot software, run time, or design time could affect the previous behavior of the bot. The design of the bot behavior could also include CE, as a change to the behavior of the bot (adding an additional click in the workflow) could be proportional to the size of the program/configuration expressing that behavior.

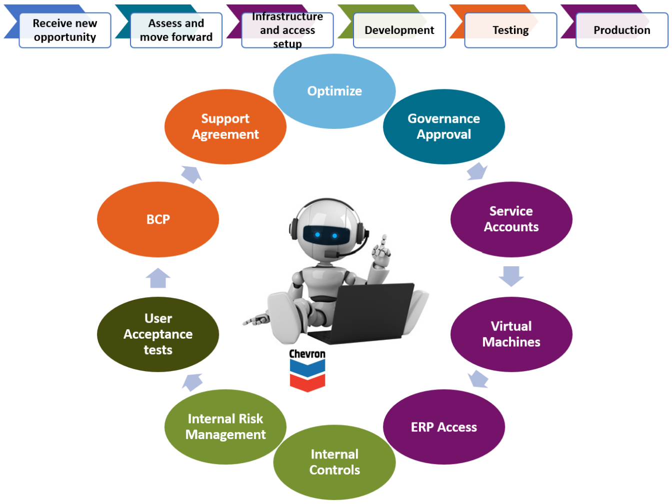

Figure 8. RPA path @ Chevron - from [21]

IV. RPA CASES

In this section, 2 international companies in the energy sector, Chevron and Engie, testify about thier experience with RPA.

A. Chevron

During Oracle Open World 2019 [21] , Carolina Barcos and Enrique Barrantes of Chevron shared their experience with RPA during the session "Robotic Process Automation: Lessons Learned". Chevron (a worldwide utilities company) has defined an RPA path (see Figure 8), which consists of six steps. During the "Process intake" step, the candidate process is being investigated for suitability for RPA. During the "Assess and move forward" step, the business case for applying RPA on the process is being presented to get governance approval. During the "Infrastructure and access setup" step, application accounts, application access, the compute resources, and service accounts are being collected. In the "Development" step, the bot rule base is developed, and the necessary internal controls are put in place to feed internal risk management. During the "Testing" step, the bot undergoes user acceptance tests. In the final "Production" step, the Business Continuity plan is set up (what if the bot fails?), and the support agreement with the business is drawn.

Chevron quickly learned that there is no such thing as a "simple process". Business typically oversimplifies their process description, and at the beginning of the RPA initiative, only a part of the actual process is known. Because of the popularity of RPA, the business is sometimes too eager to use RPA, while there may be other quick wins and low hanging fruit available to optimize the process. An essential enabler of the RPA setup process is to get risk management, internal control, and business continuity on board asap and to get their approval before go-live. In terms of development approach, Chevron goes for agile: fail fast – improve – do it again. Chevron identifies the collection of all accesses to all required systems and applications, as an attention point.

At Chevron, RPA is not seen as a local and personal productivity tool. A lot of effort goes into setting up the right RPA environment, running on virtual machines, having development, acceptance, and production bot and using scheduling and orchestration between bots. The impact of infrastructure changes, such as Windows patching, is recognized. The last but not least lessons learned is the need to be in the loop of "unexpected changes in systems," . . . which seems like a contradiction. The experience certainly comes from often encountered bot failure or incorrect behavior due to change the RPA team was not aware of.

B. Engie IT

Engie IT is part of Global Business Services at Engie (Utilities and Energy Services company). Within Engie IT, a particular group has been set up which helps the business with the setup of RPA solution and the operational maintenance of the RPA solutions. During a meeting, both the results and reflection of Section III and the Chevron case where presented. The team wholeheartedly agreed with the conclusions from Section III and has similar experiences as Chevron. Engie IT has the advantage of being part of a lot of IT initiatives within the group. When they receive a request to use RPA, the process and applications are also checked against known ongoing efforts, like consolidation or improvements at group level. This can help in the decision to go for RPA (process suited or new initiatives related to process optimization are too far away), or not (process not suited or process/application will undergo major change soon). The RPA team admits that it's sometimes hard to properly evaluate a process/application for RPA suitability. The team was seeking additional guidance. It was those remarks which triggered the creation of this paper. They mentioned one compelling use case, which was not mentioned in the studied literature. Within Engie here as some strategic initiatives regarding consolidations and moves towards SAP4HANA. They found RPA to be an excellent solution to migrate data from the current systems to the new ones. The creation and testing of special application used for the transfer of data from the current application to the new application, can be more cumbersome and expensive compared to a simple re-keying all information from the as-is application into the to-be application, by a short-lived, straight forward, never tiering bot.

V. DISCUSSION

Not every process is suitable for RPA, but still, a lot of processes, especially supporting processes, are a good candidate for RPA. Sufficient cases exist where RPA turns out to be quite profitable (just Google "RPA success stories" and/or see [22] [23] [24] for some examples). Based on the analysis done in Section III, the conclusion can be drawn that RPA is inherently unstable under change. By only looking at the UI, the origin of change and reason for change cannot be deducted. Both Chevron and Engie IT confirm that protection against unplanned change is not possible. The only kind of reasonable protection is proper logging of the bot steps and detection of which action caused failure. As shown in Section III, that may tell when a crash occurred but not to why it occurred.

The "Rule of 5" of Forrester [8] is a mechanism to make sure that the complete system does not become too big. Although not presented or described this way, the rule recognizes the CE in the system. If the system is not too big, you can still handle the CE. But as the system grows, the effect becomes larger up to the point where it no longer manageable.

RPA tool vendors already mention today the usage AI in bots, allowing them to autocorrect and thus compensate for the inherent instabilities of RPA due to change. One must be skeptical about those kind statements for multiple reasons. First, if a bot would make use of programming techniques where all screen elements are represented by concepts that are independent of the actual screen layout or usage of relevant positions, a bot could indeed be protected from cosmetic changes on the screen. Although a smart move, it has nothing to do with AI. Second, as the UI is the aggregation of all concerns, in the UI there is no data available on which a Machine Learning Algorithm could perform any kind of learning. The “cause” data is not available; only the “effect” data (not working) is. Third, even if data about the origin of the change is available, where would the training data for the AI come from? For a Machine Learning Algorithm, large quantities of data on all kinds of changes at the process, application, and infrastructure layer, plus their effect and remediation, need to be available to have a decent set of training data. Where would this data come from? For a Deep Learning Algorithm, the same restriction holds.

AI could work if it is fed by data external to the RPA digital world. But if that is the case, then it must mean that AI is able to use data from within an application and thus have access to internal data and function of the application. In those cases, one can ask the question of what would be the point of still using RPA, as all elements could be on the table to have a real application integrate with existing applications and thus perform automation via programming, even via low code programming, without having to change the existing IT landscape as well. This paper does not have the ambition to prove the statements around AI formally. What it does want to do is to provide a critical note using a reductio ad absurdum approach. The subject should be further investigated.

One of the Engie IT RPA support engineers is seeing RPA tool vendors moving toward API based interaction between applications instead of UI based interaction. This means that those vendors are moving more into the realm of BPM/A.

Besides the potential stability issues of RPA, some other perverse side effects may arise. An Enterprise Architect of a Belgium Banking and Assurance company, sees RPA becoming a blocking factor for system evolution and innovation. As the business has invested a lot of money in RPA, changes to their existing landscape directly impact their RPA investments. The business becomes reluctant to improve and innovate on their current landscape and chooses a status quo to protect their RPA investments.

VI. Conclusion

RPA is popular, certainly within the business, as it offers a fast, cheap, and non-intrusive way to boost the performance of business processes. Although some rules of thumb exist regarding which processes to choose, current studies go past the inherently unstable nature of RPA. The impact of change is reported, not the root cause. Using NS, the reasons for the impact become clear: violation of the minimum requirements for evolvability.

AI is often mentioned as a mechanism to compensate for the instabilities of RPA due to change, but additional research on this topic is required. The stability of the RPA configuring and programming methods with regards to change – add new clicks, strokes – has been left out of scope but merits additional research as well.

Acknowledgment

The authors wishe to express their gratitude to Engie IT and Chevron for sharing their experiences, and to Frederik Leemans, Bertrand Perardelle and Stefan Thys for reviewing and commenting on the paper.

References

[1] M. Kirchmer, “Robotic process automation - pragmatic solution or dangerous illusion?” 2017, URL: https://www.researchgate.net/publication/317730848 Robotic¡ _Process_Automation_-_Pragmatic_Solution_or_Dangerous_Illusion, [accessed: 2020-08-01].

[2] K. Osmundsen, J. Iden, and B. Bygstad, “Organizing Robotic Process Automation: Balancing Loose and Tight Coupling,” in Proceedings of the 52nd Hawaii International Conference on System Science, Jan 8-11, 2019,Grand Wailea, Maui, Hawaii, Januari 2019, pp. 6918–6926, URL: https://scholarspace.manoa.hawaii.edu/handle/10125/60128?mode=full, ISBN: 978-0-9981331-2-6, [accessed: 2020-08-01].

[3] S. Madakam, R. M. Holmukhe, and D. K. Jaiswal, “The future digital work force: Robotic process automation (rpa),” JISTEM-Journal of Information Systems and Technology Management, vol. 16, 2019, URL: https://www.scielo.br/scielo.php?script=sci_arttext&pid= S1807-17752019000100300, [accessed: 2020-08-01].

[4] S. Jovanović, J. Durić, and T. Šibalija, “Robotic process automation: overview and opportunities,” International Journal ”Advanced Quality”, vol. 46, no. 3-4, 2018, pp. 34–39, URL: http://journal.jusk.rs/index.php/ijaq/issue/view/12, [accessed: 2020-08-01].

[5] C. LeClain, “Use The Rule Of Five To Find The Right RPA Process,” Forrester, September 2018.

[6] B. Evelson and C. LeClain, “Look To Four Use Case Categories To Push RPA And AI Convergence,” Forrester, August 2018.

[7] C. Craig, “RPA, DPA, BPM, And DCM Platforms: The Differences You Need To Know,” Forrester, March 2019.

[8] C. LeClain, “Attended-Mode RPA: The Differences You Need To Know,” Forrester, August 2019.

[9] H. Mannaert, J. Verelst, and K. Ven, “The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability,” Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, URL:https://www.sciencedirect.com/journal/science-of-computer-programming/vol/76/issue/12, [accessed: 2020-08-01].

[10] ——, “Towards evolvable software architectures based on systems theoretic stability,” Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116, URL:https://onlinelibrary.wiley.com/toc/1097024x/2012/42/1,[accessed: 2020-08-01].

[11] P. Huysmans, G. Oorts, D. Bruyn, H. P., Mannaert, and J. Verelst, “Positioning the normalized systems theory in a design theory framework,” in Second International Symposium, BMSD 2012, July 4-6, 2012,Geneva, Switzerland. Springer, 2012, pp. 43–63, URLhttps://link.springer.com/book/10.1007/978-3-642-37478-4, [accessed: 2020-08-01].

[12] L. Ivančić, D. S. Vugec, and V. B. Vukšić, “Robotic process automation: Systematic literature review,” in Proceedings of Business Process Management: Blockchain and Central and Eastern Europe Forum, Sept 1-6, 2019, Vienna. Springer, 2019, pp. 280–295, URL: https://link.springer.com/book/10.1007/978-3-030-30429-4, [accessed: 2020-08-01.

[13] “Quote Bill Gates on automation,” 2015, URL: https://www.capgemini.com/2015/01/tempted-to-rewrite-bill-gates-rules-on-automation/, [accessed: 2020-08-01].

[14] B. Bygstad, "Generative innovation: a comparison of lightweight and heavyweight it," Journal of Information Technology, vol. 32, no. 2, 2017, pp. 180–193, URL: https://link.springer.com/journal/41265/32/2, ISSN: 0268-3962 (Print) 1466-4437 (Online), [accessed: 2020-08-01].

[15] J. Gao, S. J. van Zelst, X. Lu, and W. M. van der Aalst, "Automated robotic process automation: A self-learning approach," in Proceedings of "OTM Confederated International Conferences" On the Move to Meaningful Internet Systems, Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, October 21–25, 2019, Rhodes, Greece, ". Springer, October 2019, pp. 95–112, URL: https://link.springer.com/book/10.1007/978-3-030-33246-4), [accessed: 2020-08-01].

[16] M. Lacity and L. P. Willcocks, Robotic process automation and risk mitigation: The definitive guide. SB Publishing, 2017, ISBN:978-09-95-68-20-30.

[17] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized systems theory: from foundations for evolvable software toward a general theory for evolvable design. Koppa, 2016, ISBN: 978-90-77160-09-1.

[18] "ArchiMate 3.1 Specifications," 2019, An Open Group Standard URL: https://pubs.opengroup.org/architecture/archimate3-doc/, [accessed: 2020-08-01].

[19] D. Van Nuffel, H. Mannaert, C. De Backer, and J. Verelst, "Towards a deterministic business process modelling method based on normalized systems theory," International Journal on Advances in Software, vol. 3, no. 1-2, 2010, URL:http://www.iariajournals.org/software/tocv3n12.html, [accessed: 2020-08-01].

[20] G. Haerens, "Investigating the applicability of the normalized systems theory on it infrastructure systemst," in Workshop on Enterprise and Organizational Modeling and Simulation (EOMAS), 2018, Jun 11-12, Talinn, Estonia. Springer, June 2018, pp. 123–137, URL: https://link.springer.com/book/10.1007/978-3-030-00787-4, [accessed: 2020-08-01].

[21] "Robotic Process Automation: Lessons Learned," 2014, Presentation Oracle Open World 2019, URL: https://events.rainfocus.com/widget/oracle/oow19/catalogow19?search=Chevron, [accessed: 2020-08-01].

[22] A. Asatiani and E. Penttinen, "Turning robotic process automation into commercial success–case opuscapita," Journal of Information Technology Teaching Cases, vol. 6, no. 2, 2016, pp. 67–74, URL: https://link.springer.com/journal/41266/6/2, ISSN: 2043-8869 (Online), [accessed: 2020-08-01].

[23] S. Balasundaram, , and S. Venkatagiri, "A structured approach to implementing robotic process automation in hr," in Proceedings of the Third National Conference on Computational Intelligence NCCI 2019, 2019 Dec 6-7, Bangalore, India, vol. 1427, no. 1. IOP Publishing, 2019, p. 012006, URL: https://iopscience.iop.org/issue/1742-6596/1427/1, [accessed: 2020-08-01.

[24] W. A. Ansari, P. Diya, S. Patil, and S. Patil, "A review on robotic process automation-the future of business organizations," 2019, URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3370211, [accessed: 2020-08-01].

 ISBN: 978-1-61208-783-2