

RPA の動向と展望に対する一提言

田中 健吾^{1),2)}

- 1) 香蘭女子短期大学 情報センター
- 2) 香蘭女子短期大学 ライフプランニング総合学科

tanaka@koran.ac.jp

A Proposal for Trends and Outlook in RPA

Kengo Tanaka^{1),2)}

- 1) Information Technology Center, Koran Women's Junior College
- 2) Comprehensive Studies for Life Planning, Koran Women's Junior College

概要

RPA は人件費削減を目的に、専門家がソフトウェアロボットを大規模に導入するというスタイルが、現在の主流であるが、本稿では、非専門家が、手軽にいつでも業務に必要なロボットを RPA で作成し、日常業務に役立てるという小規模なセグメントの重要性を考察する。今後、RPA の小規模なセグメントが充実していくために、情報技術とビジネスの両方の視点から重要であると考えられる、いくつかの提言を行う。

1 はじめに

RPA (Robotic Process Automation) は 2010 年代の半ばから、日本市場にも登場し、ソフトウェアロボットやデジタルレイバーといった、人間に代わって業務を遂行してくれるロボットを想起させるキーワードで、主としてバックオフィス業務の自動化という観点から関心を集めてきた。

当初は外国製品が先行していたが、間もなく純国産の RPA がリリースされたり、OEM としての RPA も登場したりと、RPA 製品の種類や機能性も充実・多様化し、販売業者は企業での活用事例の創出と大規模導入に邁進してきた。その様な中で、紙媒体の資料をテキストデータ化し、入力するという OCR+RPA という典型的な活用事例も生まれた。最近では、機械学習を備えた OCR が登場し、手書き資料も高い精度でテキストデータ化することが可能となり、AI+RPA という事例も登場しつつある。RPA の入門的解説と活用事例に関するセミナーや書籍も日常的に目にするようになり、活況となった日本での RPA 市場は、いわゆる黎明期を通り過ぎ、次の段階へと進んでいるように見える。

RPA は働き方改革や労働人口の減少問題と関連付けられて説明されることも多い。総務省は 2018 年 9 月に「地方自治体における業務プロセス・システムの標準化及び AI・ロボティクスの活用に関する研究会 (通称：スマート自治体研究会)」を

設置し、その後、2019 年 5 月に同研究会の報告書を提出している[1]。その中で、「人口減少が深刻化しても、自治体が持続可能な形式で行政サービスを提供するため」「職員を事務業務から解放し、職員でなければならない、より価値のある業務に注力できる環境を作るため」の解決策の一つとして、AI・RPA の活用が提案されている。

地方自治体における AI・RPA の実証実験・導入状況の調査データ[2]についての記述もあり、AI の活用事例としては音声認識による議事録作成や住民からの問い合わせに対応するチャットボットの導入などが顕著であることが紹介されている。他方、RPA の活用事例は専らデータの入力・処理業務である。また、AI・RPA の導入が進んでいるのは、指定都市・中核都市等の人口が一定以上規模の自治体を中心であり、小規模の自治体では導入が進んでいないこと、AI の導入団体の大部分は実証実験段階での無償導入であること、が指摘されており、価格面・予算確保が今後の課題であるとの記述がなされている。

以上で述べたように、大幅な人件費削減を目指して、専門家が精密な設計のもとに、大規模に RPA を導入するという事例が企業でも自治体でも話題の中心となっている。しかし、RPA 製品には、「ノンコーディング」「プログラミングの知識不要」「誰でも簡単にロボットが作れる」などといった、非専門家が手軽に扱えるといったイメージのキーワ

ードが、多くの場合、伴っており、上記とは相反するような印象を受ける。実際、RPA のセミナーに参加すると、専門家による「ヒアリング」「シナリオ作成」「多数のロボットを常時稼働」などといった案内を受けることになる。

ビジネスとしては、もちろん、大きな利益が見込めるような大規模な案件が優先されることは常だと思うが、非専門家が手軽にといった RPA の趣旨は、大規模に業務を自動化するという、現在の RPA の主流の趣旨と比較しても、決して重要度が低くないと、筆者はこれまでの RPA による開発経験を通して考えている[3-5]。本稿では、その点について、RPA の技術特性や RPA と類似する技術などを考察しつつ、いくつかの提言を行いたい。

2 RPA の技術特性

この節では、RPA の優位な機能である、スクリーンスクレイピングとノンコーディング開発について、その技術特性と創造する価値について簡潔に述べたい。

2.1 スクリーンスクレイピング

RPA が持つ最も優位な機能はスクリーンスクレイピング（画面認識）であり、デスクトップや Web 画面を直接操作することができる機能である。このことは、直ちに API が無くてもアプリケーション間の連携が可能であることを意味する。即ち、API が不在で、人間が直接デスクトップ操作を行うことでデータ連携する部分を、RPA でデスクトップ上のオブジェクト操作に関するプログラミングをすることで、その操作が自動で実行可能になる。

また、上記の機能を用いることで、いわゆる Web スクレイピングも可能であり、Web サイトから有用なデータ抽出を行い、何らかのアプリケーションに記録することもできる。

スクリーンスクレイピングでデスクトップ上のオブジェクトを指定する方法は、デスクトップ上の座標指定を用いたり、あらかじめ、オブジェクトを画像認識しておき、実行段階で同一の画像のオブジェクトを探して指定したり、HTML コードの解析を行い、タグを利用することで指定したりするなどの技術に基づいている。したがって、Web アプリケーションの操作に関しては、開発側の都合で、ある日突然、仕様が変わってしまうと、たちまち機能しなくなるという事態に至ってしま

う。

2.2 ノンコーディング開発

もう一つの RPA の優位な機能として、ノンコーディング開発が挙げられる。RPA のプログラミングにおける各命令は、ビジュアル化・モジュール化され、マウス操作で命令を挿入し、その引数部分の入力を求められる仕様になっている。

ノンコーディングでの開発は、コードの記述量や、記述ミスなども軽減され、工数や開発時間などを大いに削減してくれる。また、2.1 で述べたように、RPA でデスクトップ操作をすることで、API が不在でもアプリケーション間の連携が可能になる。このことは、データ処理速度の問題はあるものの、RPA を用いることで、API に相当する機能をノンコーディングで開発していると換言してもよい。仮に API が開発可能だとしても、その開発を回避することができる。このような機能性から、開発コスト全般が大幅に削減されることが期待できる。

3 考察と提言

RPA は通常、サーバにインストールされ、同時に何体ものロボットを動作・管理できるものが主流である。他方、ライセンスに応じて機能制限されたものもあり、その中でも、RPA を端末にインストールして、ロボット 1 体はその端末のデスクトップを操作するという形態のものを、RPA と区別して RDA（Robotic Desktop Automation）と呼ぶ場合がある。

RPA と RDA を区別すべき必然性があるか否かはここでは議論しないが、1 節でも触れたように、非専門家が手軽に RPA もしくは RDA を利用できる、ということの重要性と意義について、本節では述べたい。ここで、十分注意しておきたいのは、単なる低価格設定にすべきだという主張ではないという点である。現在の主流である大規模かつ専門的な RPA の利用方法というセグメントと同程度に、専門家の設計・施工を介さずに、汎用的かつ手軽に RPA を小規模に

利用するというセグメントの重要性を考察しつつ、同セグメントが充実するために、情報技術とビジネスの両方の視点で重要だと考えられる内容について議論を進めたい。

3.1 対人件費という価格設定

RPA は人的コストを削減するという名目から、ライセンス使用料の価格設定が対人件費との比較で行われている傾向にあり、かなりの高額である。機能に応じて価格帯も異なるが、年間のライセンス使用料が数十万円から数百万円といったところであり、スモールスタートしづらい状況にある。無償での試用版も提供されているが、ほとんどの場合、期間が限定されており、ユーザーはその期間内にライセンス使用料を超えるだけの、RPA で処理するタスクとメリットを見出さなければならない。

RPA の導入段階で、一人分もしくはそれ以上の労働量に相当するタスクを揃えることが可能かどうかは、その職場環境や業務内容に大きく依存していると思われる。一般に、企業にとっては人件費が最大の支出だと言われるが、人件費の自由なコントロールが難しいという点を、ソフトウェアロボットまたはデジタルレイバーであれば、その時の企業の状況に応じて、スケールイン・スケールアウトもしくはスケールダウン・スケールアップできるというところにも、真の価値があると考えられる。スモールスタートがし易くなるためにも、近い将来、従量課金のシステムが充実していく必要があることを提案したい。

最近、マイクロソフト社が従来の Flow というサービスを拡張する形式で、Power Automate という RPA をリリースした。既存の RPA と比べると現状では機能不足な点があるかもしれないが、価格面では大幅なコストパフォーマンスを示しており、小規模・一般的な利用を想定したセグメント向けのサービスとなっている。

3.2 PEPPER のメンタリティ

PEPPER は 2014 年にソフトバンク社によって開発された感情認識ロボットであり、一般家庭

や店舗などに導入されている。筆者は RPA を利用し始めたときに、PEPPER を思い浮かべた。PEPPER の主たる役割は、家族や来客とのコミュニケーションを豊かにすることだが、RPA が、PEPPER の存在意義が意図するメンタリティと同趣旨で、日常的に手軽に利用できるようになれば、真の意味で人間の日常業務を手助けする身近なロボットになるであろうと直感した。

しかし、現状の RPA の主流は専門的かつ大規模に導入されるスタイルであり、非専門家が手軽に利用するといった、一般性や即時性、柔軟性に重きは置かれていない。誰もが必要な時に、手軽に業務に必要なロボットを RPA で作って利用できるというレベルにはかなりのギャップを感じる。このセグメントをターゲットとしたビジネスが成立するか否かは、価格設定や課金方式はもちろんであるが、それに加えて、RPA の販売業者が導入時のコンサルティングを前提にビジネスを行うという大企業向けのセグメントと切り分けられるかどうか、また、真の意味で RPA がノンコーディング開発をどこまで具現化できるか、という点にかかっていると思われる。奇しくも、PEPPER にも Choregraphe というノンコーディング開発のツールが備わっている。

3.3 自動化ツール

RPA 以外にも、処理を自動化するツールは存在する。一番身近なものは Excel のマクロである。Excel のマクロは画面操作を記録することができ、その後は、記録した処理を再生することで、Excel の操作を自動化できるお馴染みの機能である。身近さと手軽さという意味では群を抜いているが、自動化できるのは Excel の範囲内にとどまっており、他のアプリケーションとの連携はできない。

Windows に代表されるバッチ処理も OS やアプリケーションを操作することができるが、通常、GUI の操作には対応していない。他にも、Web スクレイピングに特化した自動化ツールや GUI のテストを自動化することを目的とした自動化ツールも存在する。

Excel のマクロと同様に、画面操作を記録し、それを再生する形式の自動化ツールも存在し、非専門家が利用しやすいものも一部には存在するようだが、現状では、RPA を含めた大半の自動化ツールは、そのセットアップや設定に必要な知識・技術を考慮すると、利用対象者は専門家を想定している範疇にとどまっているように思われる。もちろん、Excel のマクロにおいても、自動化できる範囲は、GUI で操作できる範囲に限定されているので、より高度な処理を自動化しようとするならば、VBA を利用することになる。

次項以降でさらに議論するが、本稿での提言をより明確にすると、RPA の利用が、ロボットを 1 体もしくは小数体という限定的・小規模な範囲内で、Excel のマクロと同様のレベルかそれに準ずる手軽さで、非専門家のエンドユーザーに日常利用されるというセグメントが実現することの重要性を主張するものである。

3.4 ノンコーディング開発の現状とプログラミングという行為のアップデート・セグメント化

RPA の優位性の一つに、ノンコーディング開発が挙げられ、「プログラミングの知識不要」というキーワードが散見される。2.2 で述べたように、RPA のプログラミングにおける各命令は、マウス操作でビジュアル化された命令を挿入し、その引数部分の入力を求めるビジュアルプログラミングが採用されている。

しかし、筆者が複数の RPA を使用した範囲内では、字義通りの「ノンコーディング開発」を具現化しているとするには、一定のギャップがあると言わざるを得ず、現状では、「ローコーディング開発」と表現するほうが適切であると解釈している。より正確に表現すると、コードを自ら記述する必要が無いという趣旨での「ノンコーディング開発」と「プログラミングの知識・経験不要」は異なるということであり、RPA での開発には、プログラミングの知識・経験は不可欠だと強く感じた。

プログラミングの専門家にとっては、おそらく、RPA のビジュアルプログラミングの開発環

境を「不要もしくは煩わしい」「どこまで複雑なロジックに耐えられるか不明」などと直ちに感じるようになるであろう。また、Excel にかかわる処理であれば、これは RPA ではなく VBA で対処すべき案件であるといった専門家の意見もよく見かける。

専門家の上記の様な意見や感想は極めて妥当であると筆者も感じるし、既存の枠組みでの開発が総合的にベンダー・ユーザーの双方にメリットがあれば、敢えて RPA を選択する必要性は何処にもない。つまり、専門家が開発する場合は、ノンコーディング開発で大幅に工数が削減できるのであれば、メリットが見込めるが、そうでない場合は RPA 以外で対応したほうが望ましい場合も多いと思われる。

上記は、あくまで専門家向けの話しに限定した内容である。RPA の開発環境を見れば、誰もがビジュアルプログラミングの代表的存在である Scratch を想起し、プログラミングという行為の平易化を予感させる。実際、RPA の開発環境では、コードがビジュアル化されていることで、コード展開を論理的な思考による認識と視覚的な認識で重層的に思考・認識する感覚を体験することができる。それに加えて、主体的にコードを記述するという行為から、命令を選択し、引数を記述するという行為へ移行していることで、コードの記述量や、記述ミスなども軽減され、開発がアシストされている感覚を抱く。

この様な「プログラミングという行為のアップデート」ともいべき事態を予感させる状況にあるが、現状は、ロジックを設定する際に、それぞれの RPA の開発言語である VBScript や Java、Visual Basic などの記述が必要になることが多く、ノンコーディング開発の実現はまだ道半ばといえる。

ノンコーディング開発を推進するためのヒントは、Windows OS や Word、Excel などのアプリケーションのここ 10 年間くらいの設計思想にあると考えている。即ち、非専門家には GUI の

分かりやすい位置で機能を選択させ、専門家に必要な高度な機能は、敢えて表に出さないというポリシーである。Excelで論理演算や比較演算などを用いる際も、演算子の形式ではなく言語化されている場合がある。演算子を言語化することで、その解釈に曖昧さが生じてしまう事態も予測されるが、それは、最初にプログラミングに接するのが、従来のプログラミング言語のスタイルだからであり、徹底してノンコーディング開発を追求したビジュアルプログラミングに最初から触れることになれば、違った結果になると期待される。

また、専門家が必要とする複雑なロジックに耐えうるだけのノンコーディング開発が実現できるかという課題については、すべてをノンコーディング化する必要はなく、使用頻度の低い命令や機能についてはコードを記述する形式で残すなど、専門家向けと非専門家向けとに、大きくRPAのセグメント分けをすることを提案したい。非専門家向けにノンコーディングを追求した開発環境を持つミニマムなRPAもしくはRDAが登場し、そのようなセグメントのビジネスが成立することが、真の意味で人間の日常業務を手助けする身近なロボットが存在する未来をもたらすことに繋がると思われる。

3.5 Excel というアプリケーションの立ち位置とRPA

Excelに焦点を当てることで、RPAの今後の展望について考察してみたい。誰もが知っているExcelというアプリケーションを再考してみると、幾分、特異な存在だということに気付く。

Wordは、初めて手にする人でも文字は打てるし、直感的に最低限必要な作業は行える。他方、Excelは、初めての人が意義を感じられる処理ができるようになるには、一定の学習が必要になる。また、繰り返し処理は利用できないが、関数の機能は充実しており、プログラミング的要素が含まれている。この様な直感的にすぐ使用できないアプリケーションが、現在の様に浸透しているには、その有用性と信頼性が共通認識

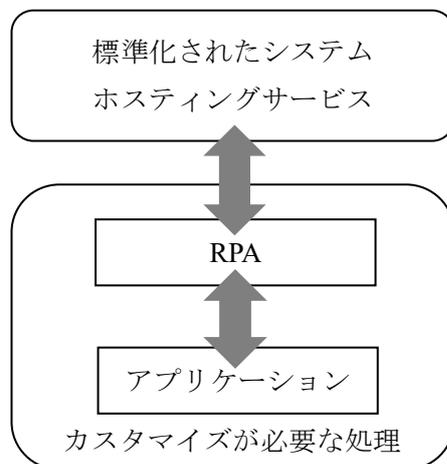


図 1 カスタマイズ問題へのRPAでの対応

されていることが最大の理由だが、そこに至るまでには、学校教育の果たす役割が大きいと思われる。実際、日本でも多くの高等学校や高等教育機関が、Excelを学ぶ機会を提供している。

3.4で提案した、非専門家向けにノンコーディングを追求した開発環境を持つミニマムなRPAもしくはRDAを、最初のプログラミング教育で利用することを提案したい。これまで、プログラミングを必要としない人にとって、プログラミング教育は教養教育という側面が強かったが、この提案が実現されると、今後は手軽にソフトウェアロボットを使用する実務手段としての教育に移行することになる。そして、従来のプログラミング教育はその後の専門的な教育に位置付けられることになる。

3.6 APIとしてのRPAとカスタマイズ問題の解決策

2.1で述べたように、現在のRPAの最も優位な機能性は、スクリーンスクレイピングによるデスクトップ操作であり、APIが不在のアプリケーションに対しても、直接、GUIを操作することにより、連携が可能になる。

非専門家向けのミニマムなRPAもしくはRDAが日常的に利用されるようになると、アプリケーション間の連携が容易になり、エンドユーザーのアプリケーションの利用方法にも変化が生じることが期待される。

ホスティングサービスならびにSaaSの利用

は、様々なコストを削減してくれるが、装備されていない機能を追加することは容易ではない。処理速度さえ問題にならないければ、RPA を用いることで、不足している機能を図 1 の様に補完することができる。その際、RPA は API としての役割を果たして、ホスティングサービスからデータを受け取り、RPA の内部で処理する場合もあれば、別のアプリケーションと連携して処理を行う場合も想定される。

文献[1]では、自治体現場での要望による個別のカスタマイズが開発費を押し上げている問題と、それを回避するためにシステムの標準化を想定した開発の重要性に関する指摘が、文献中の各所で、再三、言及されている。このカスタマイズ問題も、図 1 の通り、RPA に API の役割を担わせることで解決出来ることが期待される。その際も、RPA のノンコーディング開発が高いレベルで実現され、真の意味で人間の日常業務を手助けするロボットになり得ることが肝要である。

4 おわりに

RPA という日常業務に役立つソフトウェアロボットが登場したことにより、非専門家にもプログラミングを習得する必然性が生まれた。そして、非専門家がプログラミング可能なように、ノンコーディング開発も進んでおり、プログラミングという行為が非専門家にも可能になるようにアップデートされるか、やはり専門家が行う行為の範疇に留まるのか、いまその分水嶺の付近にいる時期だと思われる。

3.5 で提案したように、ノンコーディングが追求された開発環境を持つ RPA や RDA が最初のプログラミング教育に位置づけられたとする。その結果、Excel のマクロを利用するのと同レベルで、多くの人が RPA でノンコーディング開発を行えるようになったと仮定しよう。その次に起こり得るハードルは、かつての Excel と同様に、その処理された結果が信頼されるかという点であろう。Excel が利用されだした当初、Excel

で処理した計算結果を信用できずに、電卓で計算するという話があった。非専門家が作ったロボットが処理した結果が、Excel での処理結果と同様のレベルで信用されるには、学校教育での充実などを通して、一定の期間を要するものと考えられる。また、自治体現場でその処理が認められるには、さらに時間を要するかもしれない。

文部科学省が昨今、提唱している文理を問わず、すべての生徒・学生に AI・データサイエンス教育の機会を提供するという趣旨には、本稿の RPA もしくは RDA を用いたプログラミングの入門教育は、ノンコーディング開発による学習面のハードルの低さと、RPA が近い将来、AI の機能が搭載されるというロードマップが予想されていることより、親和性が高いと思われる。そのためにも、RPA を非専門家が手軽に利用できるというセグメントのビジネスが、Excel に準ずる規模で展開され、RPA が API の役割を果たすことで、多様なサービスやアプリケーションの連携が生まれることを強く期待したい。

参考文献

- [1] 地方自治体における業務プロセス・システムの標準化 及び AI・ロボティクスの活用に関する研究会 報告書、
https://www.soumu.go.jp/main_content/000620983.pdf、2019 年 5 月
- [2] 地方自治体における AI・RPA の実証実験・導入状況等調査、総務省自治行政局行政経営支援室、
https://www.soumu.go.jp/main_content/000624150.pdf、2019 年 5 月
- [3] RPA と G Suite の連携による学生の記述・投票共有システム、第 18 回情報科学技術フォーラム、第 4 分冊 P 79-82、2019 年
- [4] RPA の教育分野への応用－BizRobo! と G Suite の連携による著作権教育－、日本教育工学会研究報告集 教育方法・授業改善/一般、2019 年
- [5] RPA の学生支援における活用－BizRobo! による就職上サイトのスクレイピング－、大学 ICT 推進協議会 2019、2019 年