



Robotic Process Automation

R P A 概 説

総集編 ① - ⑥

躍進を続けるソフトウェアロボット
R P Aによる働き方改革は次のステージへ

RPA 概説 目次

RPA 概説① RPA って知っていますか？ … 3

- RPA とは？
- 産業用ロボットとの違い
- RPA 導入のメリット
- Digital labor の利点
- AI への期待
- 機械学習とディープラーニング
- RPA の特徴
- RRPA の適用分野
- RPRA による業務改革
- RRPA 対象業務の洗い出し
- RRPA の導入コスト
- RRPA の仕組み
- RRPA と BPO
- RRPA によるビジネスチャンス

RPA 概説② RPA 導入のために知っておきたい豆知識 … 11

- RPA はデジタル労働者
- 自動化の手法
- テンプレート型 RPA
- 開発型の RPA
- 2つの型の比較
- 代表的な RPA ベンダー
- その他の RPA ベンダー
- 国内の RPA ベンダー
- BPA と RPA
- RPA の対象業務選定のポイント
- テンプレート型の選定ポイント
- 開発型を選んだ方がよい場合
- AI を見据えた拡張性
- RPA 導入のためのポイント

RPA 概説③ ソフトウェアロボットの作り方 … 19

- ソフトウェアロボットを作るには
- ROBOWARE - フレームワーク
- ROBOWARE - API
- フレームワークとライブラリの違い
- ROBOWARE の導入
- サンプルスクリプトを選ぶ
- RBF API プログラム開発 リレー通信
- API の種類
- ROBOWARE の画面表示認識
- ROBOWARE アナライザー
- ROBOWARE プログラム管理
- Job コマンドの新規登録
- ソフトウェアロボットの実行方法
- ROBOWARE 構成例

RPA 概説④ AI 搭載を見据えた RPA の実装方法とは？ … 27

- 働き方改革の要は RPA
- RPA を社内でも進化させる
- RPA で期待される AI
- 業務に必要な AI とは？
- AI って何ですか？
- 弱い AI と強い AI
- 業務の分類と AI 技術
- ルールベースの AI
- 推論エンジンと知識ベース
- 機械学習の発展
- ディープラーニングとビッグデータ
- AI サービスと RPA
- AI 搭載 RPA の今後
- RPA による働き方改革

RPA 概説⑤ RPA2.0 を加速するソフトウェアロボット開発 … 35

- RPA 2.0 とは？
- 企業内でのロボット開発競争
- ツールとしてのロボット
- RPA ブームの問題点
- 間違っていたロボット化のアプローチ
- プログラミング技能の重要性
- AI への過大な期待
- 顧客満足度と社員満足度
- 時短を実現するために
- 単独ロボットから連携ロボットへ
- 冗長化とバックアップ
- 多様性への対応
- IoT とのネットワーク連携
- RPA の未来

RPA 概説⑥ RPA の盲点 IT ガバナンスの重要性 … 43

- ガバナンスの必要性
- ロボット運用の不安
- ソフトウェアロボット稼働の種類
- 誰がロボットを管理するのか？
- ロボット管理の進め方
- 運用リスクのマネジメント
- RPA とセキュリティ
- パスワード設定のリスク
- エラー処理の重要性
- ディザスタリカバリ
- バージョンアップへの対応
- 誰がロボットを作成すべきか？
- ソフトウェアロボットの責任所在
- ガバナンスと企業の将来

RPA って知っていますか？

2017年01月31日公開

<https://www.esector.co.jp/special/rpa/rpa.html>



RPA とは？

近年、注目を集めている RPA は、Robotic Process Automation の頭文字です。つまり、ソフトウェアロボットが、業務プロセスを自動化することを指します。

複数のアプリケーションを連携して操作したり、表示した画面の内容を確認して入力する作業など、今まで人手で行っていた事務作業を、ソフトウェアロボットが代行してくれます。

事務作業の自動化の仕組みをプログラミングをすることによって作成する場合、異なるアプリケーションをどのように連携するかが難題でした。その場合は、該当のアプリケーションの API (Application Programming Interface) を公開してもらうか、あるいは新たに作成してアプリケーション間でデータのやり取りをする必要があるため、自動化の推進が困難でした。また、そうした連携プログラムは一度作成してしまうと、アプリケーションのバージョンアップ時に確認や修正が大変でした。

それに対し、RPA は、使用するアプリケーションは変更せず、人が行う PC 操作等のオペレーションをソフトウェアロボットがそのまま同じように自動で行います。たとえば、WEB ページに表示された画面の特定の文字を探して、関連するデータをコピーして、社内システムを自動的に立ち上げそのデータを入力したりできます。

RPA は、主に定型業務はもちろん、ワークフローが定まっており手順書に基づいて処理されている事務作業などを中心にソフトウェアロボットが業務操作を自動化します。



産業用ロボットとの違い

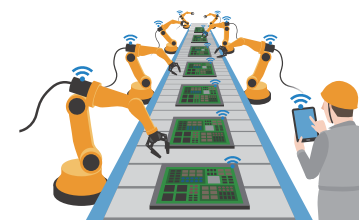
RPA は、多くの場合ロボット専用のハードウェアを必要とせず、既存システムの PC などソフトウェアロボットを導入し、カスタマイズして動きます。

ソフトウェアロボットは、経年劣化の心配はなく、機械である産業用ロボットのように油を注入したり、長く保つためにきれいに磨いたりする必要はありません。

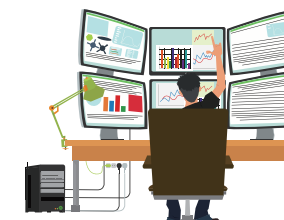
産業用ロボットが、手や足など身体のある部分の動きを代行してくれるのに対し、RPA は、主にキーボードやマウスクリックなどの PC 操作を代行してくれます。データから判断することが必要な作業も補ってくれるので、産業用ロボットが重労働に強いのに対し、RPA は頭脳労働に強いといえます。

つまり、ブルーカラーの人達を楽にしたのが産業用ロボットであるのに対し、ホワイトカラーの人達を楽にしてくれるのが、RPA のソフトウェアロボットです。今までも、ERP (Enterprise Resources Planning) など業務アプリケーションの導入で、かなりシステムの効率化は実現されてきましたが、ヘルプデスクや、多種多様なシステムの監視など、複雑なオペレーションについてはまだまだ人力に頼る部分が多くあります。

事務系のオペレーションであっても、単純作業や繰り返し行われる作業は、ソフトウェアでロボット化が可能です。今まで、無理だと諦めていた事務作業も、RPA によって自動化が期待できます。これにより、ホワイトカラーの人達も、日々の単純なルーチンワークから解放される時代がやって来しました。



産業用ロボット



RPA

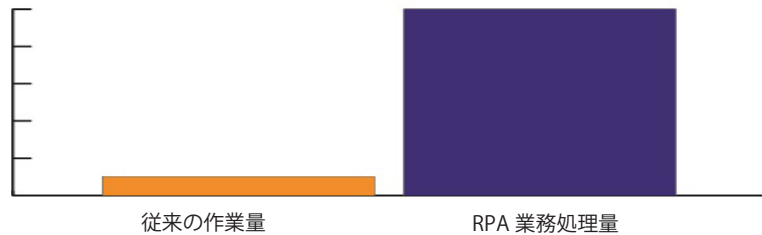
RPA 導入のメリット

RPA の導入により、企業や組織は多大なメリットが期待できます。多くの場合、投資対効果が非常に顕著に現れます。事務作業において、大幅な業務改善が図られ、企業の発展に大いに貢献します。

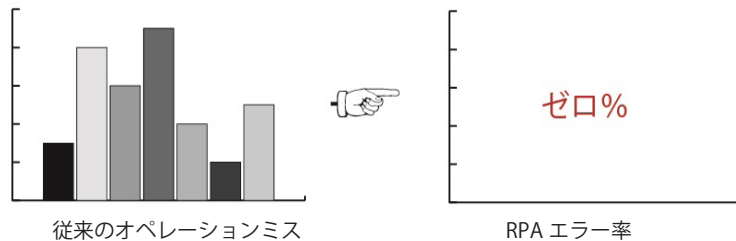
① コストメリット



② 生産性向上



③ 品質向上



Digital labor の利点

RPA のシステム、あるいはそれを構成するソフトウェアロボットは、Digital labor (デジタル労働者) と呼ばれることがあります。

Digital labor は、産業界で活躍するロボットとは別にカテゴリ分けされ、事務的業務を人に代わって作業をするバーチャルな労働者として、無限の力を秘めた労働力を提供してくれます。それでは、人間の労働者と比べてみましょう。

Digital labor の利点 (人間との比較)

- ・とにかく仕事が速い
- ・操作が正確
- ・休憩不要、有給休暇不要
- ・残業、部署異動、業務変更可能
- ・労働組合不要
- ・遅刻・早退なし
- ・不平不満を言わない
- ・パワハラ、セクハラ問題なし
- ・社内恋愛しない
- ・中元・歳暮・年賀状不要
- ・敬語、根回し不要
- ・給与・ボーナス・退職金不要
- ・昇進・昇格不要
- ・時間外手当不要
- ・通勤手当・家族手当不要
- ・社会保険・厚生年金不要
- ・指導、業績査定不要
- ・報告・連絡・相談不要
- ・社員旅行、福利厚生不要
- ・コミュニケーション不要
- ・無駄口ゼロ
- ・飲食・つきあい不要 など…

RPA は、採用するシステムにもよりますが、導入費、維持費を差し引いても人件費よりかなりコストメリットの効果が出るのが通常です。

残業を強制しても、不平不満もなく、残業費も請求されません。処理も圧倒的に早く正確なので無敵の労働力だといえます。逆を言えば、ソフトウェアロボットに比べ、人を雇う場合には上記のようにこんなにたくさんのリスクを考慮しなければいけなかったということです。

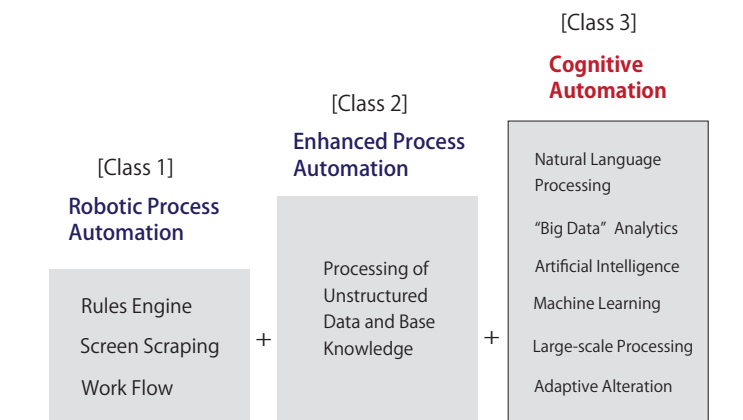
また、今までと同じ仕事ばかりでなく、近い将来、AI (人工知能) も組み込んだシステムにすれば、膨大な過去データを基に判断してくれるので、経験豊富な社員よりも優秀な成果を出す働きぶりも期待できます。

AI への期待

Robotics という言葉から、RPA にはいつも AI (人工知能) が話題になります。ホワイトカラーの仕事には、過去のデータを基に会社の継続・発展のために最良の予測・判断が必要な業務もあります。そのような仕事は、単純作業のルール化とは異なり、膨大な量のビッグデータを基に分析を要するので、RPA で簡単に自動化できるものではありません。

近年、AI に関連して、Cognitive (認知) という言葉が盛んに使用されるようになって来ました。AI 機能を持ったソフトウェアロボットは、ディープラーニング (深層学習) によって自ら学習し、膨大なデータを基に臨機応変な回答を出すということが可能になりました。これにより、天候に応じた販売予測や、生産指示など、人の力ではとても無理な量のデータ分析による対応ができるようになります。この RPA の最終進化形は、「Cognitive Automation」と言われています。

RPA の次の段階で非構造化データの解析を可能にし、3 番目の「Cognitive Automation」で、人工知能により機械学習などを利用して言語解析などが可能になります。一部では、すでに実用化が始まっていますが、今後この分野の発展によりワークスタイルがかなり変化するであろうと予想されます。このソフトウェアロボットに対し、Cognitive を実現し推し進めているキーワードが「ディープラーニング」です。



[参考: KPMG Bots in the back office - November 2015]

機械学習とディープラーニング

グーグルが 4 億ドルで買収した人工知能の会社 (DeepMind 社) の「AlphaGo」が、2016 年 3 月に囲碁の世界王者に 4 勝 1 敗と圧勝したニュースで、ディープラーニング (深層学習) という言葉が一気に広がりました。ディープラーニングは、従来から使用されていた機械学習と何か違うのでしょうか？

コンピュータにデータを処理させるためには、プログラムが必要です。従来のプログラムは、異常終了しないようにあらゆるケースを想定してプログラミング (プログラミング言語を使ってコンピュータに指示を与える) をします。それに対して、機械学習は、機械学習アルゴリズムを流し込むだけでプログラミングをする必要はありません。機械学習は、たとえば教師あり学習の場合、特定アルゴリズムの機械学習モデルに対して正解を示す教師データを大量に入力して、期待する結果が出るようになるまで、人がパラメータのチューニングをします。これに対して、ディープラーニングは、自ら学習してこのチューニングさえも不要になります。

脳機能に見られるいくつかの特性を、計算機上のシミュレーションによって表現した「ニューラルネットワーク」の階層が深いものを用いた機械学習が、ディープラーニングです。この方法で大量データを学習させるには、高度な演算能力ととてつもなく多くの処理時間が必要ですが、GPU などのコンピュータ技術の向上のおかげで、近年になってディープラーニングが実用化できました。現在ディープラーニングは、主に画像認識、音声認識、自然言語の分野で研究が進められています。

IBM 社の Watson や、IPsoft 社の Amelia のように、Cognitive system として RPA 分野での活躍が始まっています。



RPA の特徴

今注目されている RPA は、従来ロボット化が無理だと思われていたバックオフィス（間接部門）の仕事を自動化できる仕組みとして、あらゆる業種で期待されています。そこで、典型的な RPA は、どのような特徴があるのでしょうか。

- ① キーボード操作を自動化できる
- ② マウス操作を自動化できる
- ③ 画面上に表示された文字を判別して取り込める
- ④ 画面の図形や文字の色などの属性が判別できる
- ⑤ 多種多様なアプリケーションを起動、終了できる（ID やパスワードの自動入力）
- ⑥ スケジュールされた日時に実行できる
- ⑦ カスタマイズが簡単
- ⑧ 業務の手順変更柔軟に対応できる
- ⑨ 異なるアプリケーション間のデータの受け渡しができる
- ⑩ 離れたところからの遠隔操作ができる
- ⑪ 複数 PC の制御ができる
- ⑫ ワークフローや手順書通りの操作ができる
- ⑬ エラー処理など、条件分岐した処理ができる
- ⑭ 過去のデータを基に分析できる

RPA は、特定の業務に特化したアプリケーションではなく、簡単にカスタマイズできるため日々変化するビジネス要求への業務変更に対し、迅速に対応できます。コストメリットを気にしながら、スモールスタートで徐々に適用範囲を拡大していくことも可能です。



RPA の適用分野

RPA の適用分野は、あらゆる業種の様々な業務で活躍できるため特定することができません。但し、RPA の特徴から即座に効果が期待できる業務の例をあげてみます。

クラス 1 Robotic Process Automation

- ・データ入力や、複数アプリケーションの連携が必要な単純作業の定型業務
人事・経理・総務・情報システムなどの間接部門（バックオフィス）の事務・管理業務
販売管理や経費処理、アプリケーションをまたがった入力処理など

クラス 2 Enhanced Process Automation

- ・構造化されていないデータの収集や分析が必要な業務
セキュリティログの分析、様々な要因を加味した売上予測、Web のレコメンド広告など、多種のデータを基に分析を自動化する処理

クラス 3 Cognitive Automation

- ・大量のデータを基に学習して最良の判断が必要な業務
ヘルプデスクや、季節や天候に左右される仕入れ管理、経済情勢を加味した経営判断など、人間の能力では不可能と思われる膨大なデータに基づく予測をする業務

クラスが高くなるほど、導入金額は高くなりますが、その分人間が出来ないあるいは不得手な業務が可能となり、効果も絶大です。RPA が得意なのは、繰り返し作業が多い単純労働の分野ですが、AI 機能を取り入れることにより、大量の情報を基にした分析や、未来予測について経験豊富なスペシャリストが担当する業務分野までカバーします。



RPA による業務改革

RPA が人間の仕事を代行してしまうと、その分野で従事していた人の仕事がなくなり失業者が増えると心配する声があります。

しかしながら、過去の歴史を見ても、仕事は無くなりません。人間は成長する欲があるので新たな職業を創造し生活を豊かにしていきます。

100 年前、農業などに従事していた日本の第一次産業の割合は 50% を超えていたのに、現在では 10 分の 1 の 5% ほどになってしまいました。農機具の機械化や海外輸入による影響があるにせよ、代わりに第三次産業の割合が増え、今ではクラウドサービスなど 100 年前には存在すらしなかった職業がいっぱいあります。それ故、ロボット化による失業を過剰に心配する必要はありません。

ヘルプ業務にチャットボットによる自動対応を導入した事例では、担当者が一人ひとりの顧客に対応する時間が増え、満足度がアップして売上にもかなり貢献しているそうです。ソフトウェアロボットにでもできる仕事は出来るだけ任せてしまった方が、担当者はより高度な仕事に従事できます。

RPA によって代行される業務の担当者は、その業務を知り尽くしているので RPA の構築やメンテナンスをする仕事が適任です。RPA は、プログラミング能力がなくても構築できるので心配りません。

はじめて家庭に洗濯機や掃除機が入ってきた頃を、思い浮かべてみてください。家庭の主婦は、自分たちの仕事が取られるというマイナスイメージではなく、洗濯や掃除が楽になり時間が節約できて余暇が増えると喜んでいました。

RPA によって、担当者がする単純作業が減れば時間に余裕ができ、社会問題となっている残業時間も減らせます。生産性が上がれば、事業拡大も可能なので、新設した業務への異動も可能になります。

RPA のおかげで、今流行りのテレワーク（在宅勤務）も活気づくことでしょう。セキュリティが確保できれば、家からスマホでソフトウェアロボットの仕事を監視するだけで、給料がもらえる時代が来るかもしれません。

AI を組み込んだ RPA が当たり前の時代が来ると、ソフトウェアロボットが、過去のデータを基に最適な判断をしてくれるので、過去の実績と経験を評価されて高い給料をもらっていた管理職や上司はお払い箱になるかもしれません。むしろ、現場で事務処理をしていた担当者の方が、業務を分かっている RPA をおもりできるので、報酬が高くなる可能性さえあります。

RPA により、業務スタイルの変革が求められる時代になるでしょう。



RPA 対象業務の洗い出し

RPA の導入が、自社にとって有効かどうかバックオフィスの業務を洗い出してみることをお勧めします。

対象にできる事務作業は、たくさんあるはずです。今 RPA を適用すべき業務かどうかは、たとえば、下記のような表にまとめると、優先順位付けや、ざっくりとした効果算定が可能になります。

(業務洗い出しの例)

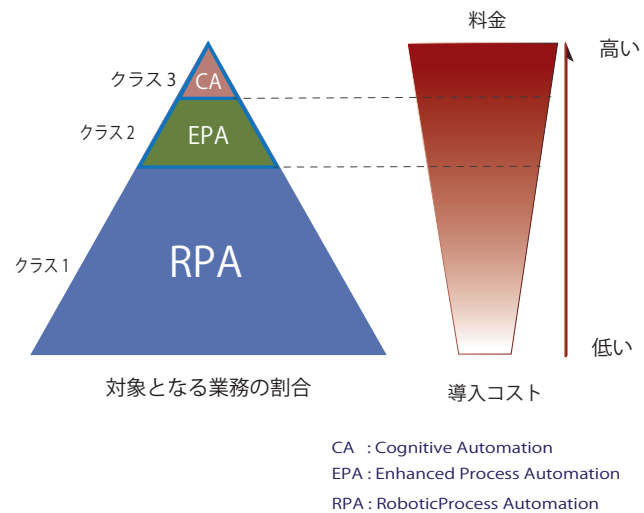
	業務A	業務B	業務C	業務D
業務目的	経費精算	売上集計	Web解析	ログ収集・解析
担当部署	経理部	営業部	マーケティング部	情報システム部
使用機器	PC3台	PC300台,iPad300台	2台	25台
使用者	3名	300名	3名	20名
使用日時・頻度	毎日	毎日	毎日	毎日
処理時間	平均2時間/日	延べ30分/日	延べ1時間/日	24時間/日
使用環境	Windows,Excel他	Win, iOS,Excel,Web	Win,Web	Win,Linux,サーバ
操作マニュアル・手順書	あり	あり	あり	あり(一部なし)
単純作業の割合	90%	60%	40%	60%
関連作業	仮払い、立替金管理	仕入れ管理	Webページ更新	アラートシステム
ケース分岐判断	あり	あり	なし	あり
影響度	2	4	1	4
コスト削減ニース	3	4	3	3
パフォーマンス改善ニース	4	3	2	4
プロセスボリューム	3	3	4	4
問題点	紙の領収書あり	セキュリティ	改善点の判断	複数アプリ
改定リスク	2	3	1	4
現状作業見積り人件費	2.0万円/月	100万円/月	30万円/月	150万円/月
変更後予定維持費用	5万円/月	30万円/月	20万円/月	40万円/月
改善しやすさ	3	2	4	4
総合判定	3	2	1	4

表にまとめる時のポイントは、業務を変更するリスクも十分考慮に入れながら、その業務へのインパクトを数値化して見ることです。関係者が多いほど、改定に反対する人も多くなります。また、もし RPA に不具合があり急遽手作業をするケースも想定して、問題点も洗い出します。

RPA の導入コスト

RPA を検討する際に、Robotic という言葉が、AI（人工知能）を搭載していることが前提となっているように勘違いしてしまうのですが、プロセスを自動化する部分は AI が無くても構築できる業務が多く存在します。

確かに、AI が活躍する事例は、画期的で脚光を浴びるのはわかりますが、実際のところディープラーニングを活かすには、その学習の基となる大量のデータが必要です。また、その学習をするには非常に多くの時間を要するため、かなりハイスペックの GPU 等を搭載した高額なマシンが必要になります。この AI の本格導入は、まだまだコストも高額になり、コンサルだけでも数千万円レベルとなってしまいます。



現実的には、3番目のクラスの「Cognitive Automation」に該当するディープラーニングを利用した仕組みを検討できるのは、ビッグデータを扱う大企業レベルの大きな組織だけです。クラス2の非構造化データの分析を自動化する場合においても、かなりの投資が必要です。実際に洗い出してみるとわかりますが、プロセスを自動化したい業務の大半は、ビッグデータには関係なく AI に頼らなくても自動化できそうな繰り返しの操作や単純作業がほとんどです。こうした RPA は、数百万円ほどで割と低価格で導入できるものが多くあります。

RPA の仕組み

近年 RPA が人気の理由は、プログラミングをして個々にアプリケーションを作成しなくても、簡単な設定で人間の操作を代行してくれるからです。

しかしながら、操作する手順や条件などは、RPA といえども事前にどのような動きをするかを詳細に設定しておく必要があります。典型的な RPA のシステムでは、パラメータで指定したり、画面操作を登録することでワークフローを作成でき、複雑な部分は独自の方法により設定を行うというものが多くあります。

業務のワークフローは、会社によって千差万別なため、RPA はカスタマイズ無しではほぼ使用できません。業務変更のたびにルール設定が面倒であると、個別にプログラミングするのと手間が変わらなくなってしまいます。また、担当者が替わるたびに導入した RPA 用の設定方法を覚えるのはたいへんです。その教育や講習会にも、費用がかかるかもしれません。それが面倒だからといって、結局業務のフローが変わるたびに業者にカスタマイズを依頼していたのでは、コストもかさみ迅速な変更も難しくなります。

現実問題として、この RPA に業務操作を設定する部分がどれだけ楽に汎用的に作られているかで、RPA のシステムの導入費用、構築費用が違ってきます。費用がかかりすぎると結局コストメリットが出せない場合もあります。そのため、RPA には、導入対象となる環境を限定しているものが数多くあります。

このような中、RPA へのルール設定をする言語として、Ruby や JAVA、PHP、C# などの汎用的なプログラミング言語を使用する方法があります。仕組みは、システムを作成しやすいフレームワークになっていて、Ruby などで記述し操作手順を組み込むのですが、コーディングが難しい部分は事前に数多くの API が用意されているというものです。この APIのおかげで、IF などの基本的な記述を覚えれば、初心者でも構築できるようになっています。

この方式のメリットは、新たに RPA ツールの設定方法を覚えるための講習会に行かなくても、汎用的な言語なので周りに知っている人も多く、ネットで参考となる資料がたくさん検索でき、利用するのが容易だということです。一度構築してしまえば、業務担当の人が直接修正も可能なので、試行錯誤しながら仕組みを作ることができる柔軟性もあります。何より、この方式であれば汎用性があり、コスト面ではかなり安く導入できます。



RPA と BPO

RPA の導入が活発になると、BPO（ビジネスプロセスアウトソーシング）が衰退するといわれていますが、今後はどのなるのでしょうか？

BPO は、自社の業務プロセスを外部企業に委託することを指しますが、社内の人的リソースが足りない場合など、社外へ業務をアウトソーシングすることにより、社内での作業を簡素化できるため近年需要が高まっております。BPO の種類によっては、中国やベトナムなど海外の安い人件費のメリットを生かしたサービスもあります。

確かに RPA によって作業プロセスの部分の負荷を軽減できれば、わざわざ業務をアウトソースする必要はなくなるようにも思えます。しかしながら、すでに BPO を利用している場合、わざわざ社内に業務を戻すのも手間とコストがかかるので、そんなに大きくコストメリットが見いだせない可能性もあります。

一方 BPO を提供している会社の方は、今後価格競争が激化すると思われるので、積極的にサービスの中に RPA を取り入れて行くでしょう。一早く RPA の仕組みを導入したデータセンターが、BPO サービスで特色を出すことができ、価格的に優位に立つことができます。クラウドサービスとして、手軽に RPA を提供するビジネスモデルも考えられます。

ユーザ側としては、BPO サービスもさらに安くなるのであれば、業務プロセスのコスト削減の方法について選択肢が増えます。RPA との比較は、コスト面よりも、今後の業務改善を外部に任せてしまうのか、自社内で完結し柔軟で且つ迅速な対応が取れるようにするのが、重要なポイントとなります。

AI の分野でも、ディープラーニングのシステムを自社で構築するには、かなり高パフォーマンスなマシンが必要になるため、IBMをはじめ、MicroSoft や Amazon などクラウドサービスとして、比較的安価に AI のサービスを提供する企業も増えて来ました。



RPA によるビジネスチャンス

RPA が企業に浸透すると、様々なビジネスチャンスが生まれます。

他社に先んじて業務効率をアップさせ、戦略的な部隊に増員なく人を配置できる環境を作るには、RPA の導入がかなりインパクトがあります。

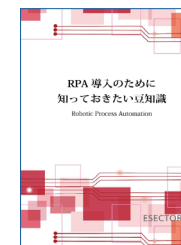
- ① 業務量増大に対応
- ② 時短、品質向上による顧客満足度の向上
- ③ コスト削減による投資金額の増加
- ④ 人材不足の解消
- ⑤ 価格競争力の増加
- ⑥ 需要の変化への迅速な業務変更
- ⑦ モチベーションの向上



今まで無理だと思われていたホワイトカラーの業務にも、ソフトウェアロボットを活用して自動化すべき時代となりました。AI 技術を利用すれば、ディープラーニングによるロボットの自己学習、自己判断により、人の手をまったく借りずとも最良な答えが出せるようになり、将来ロボットが人間の暮らしを豊かにします。

RPA は、段階的な導入が可能です。まずは、自社の事務作業で自動化できる部分を見直し、検討を始めてみてはいかがでしょうか？

RPA 導入のために 知っておきたい豆知識



2017年06月13日公開

<https://www.esector.co.jp/special/rpa/rpa2.html>

RPA はデジタル労働者

近年日本においても RPA (Robotic Process Automation) は、様々なメディアにも取り上げられ、バックオフィス業務等の自動化でとても注目されています。

デジタル労働者と呼ばれるソフトウェアロボットが行なう事務作業の代行が、産業革命時の機械化に匹敵するインパクトがあり、ホワイトカラーの働き方改革として多くの企業で導入検討が始まりました。

RPA はありとあらゆる業種で、デスクワークを中心に人手で行っていた PC 操作等をソフトウェアロボットが代行します。無理だと諦めていた複数のアプリケーションをまたがってデータ連携が必要な業務でさえ、オペレーションの自動化が可能になります。

たとえば、全社員の交通費を精算するために、経理担当者が検算して集計する処理の負担を無くす場合の想定をしてみてください。

もし、効率化のために交通費精算のクラウドシステムを導入したとすれば、そのシステムの仕様に合わせ全社員が Web 入力をする手順に変更し、レポートもクラウドシステムから出す必要があります。全体的には効率化は上がりますが、全社員が使い方を覚え、運用を変更しなければいけません。

一方これを RPA で実現すれば、社員が入力する交通費精算のフォーマットは変更せず、そのファイルをソフトウェアロボットが自動で読み込み、Web の乗換案内等のシステムを使って交通費の検算をしてくれます。部署ごとに集計していたエクセルシートもまったく同じフォーマットでソフトウェアロボットが作成します。つまり、経理担当者がしていた操作を代行するだけで、あらたに別の操作を覚える必要もなく、社員の交通費精算のフローの変更も無しに RPA を導入するだけで、経理担当者のオペレーションの自動化が実現できます。

この例でわかる通り、導入後でも容易に業務に適用出来ることが人気の要因になっています。



自動化の手法

オペレーションを自動化するための手法の代表的なものは、ルールベース、マクロ、スクリプト、API です。

ルールベース

処理手順をルールとして登録
If 条件 then else

マクロ

プログラムの複数の命令を
実行する順番に定義したもの

スクリプト

機械語への翻訳を必要とせずに
実行できる簡易式プログラム

A P I

簡潔にプログラムできるように
設定されたインターフェース

コンピュータに仕事をさせるためには、最終的には機械語にして指示が必要です。そのため、自動化する場合も作成されたアプリケーションをいつ、どのような条件で、どのように動かすかをプログラミングによって指示する仕組みが必要です。単純なものであれば、処理手順をルール化して実行すればよいのですが、複数の命令をセットにして定義したり、スクリプトによっていろいろな条件を定義してバッチ処理のように実行する方法が分かりやすく簡単です。それでも、定義できないような複雑な処理は、プログラミングを簡単にするために API を使用する方法もあります。基本的にはこれらの考え方をを組み合わせることによって、多くの RPA のシステムが構成されています。

テンプレート型 RPA

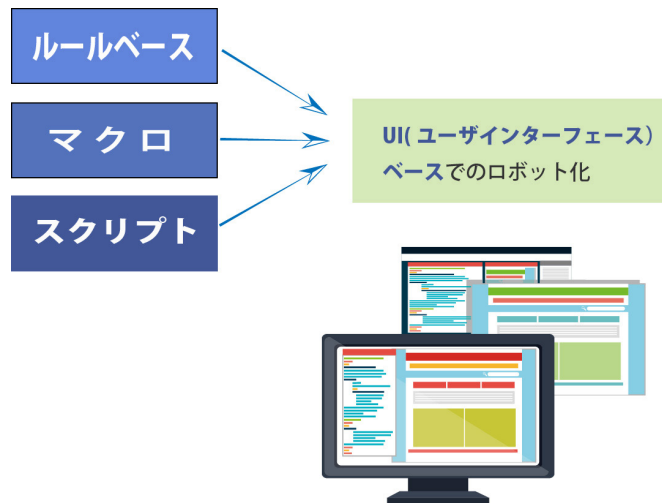
典型的な RPA ツールは、ルールベース、マクロ、スクリプトについて UI(ユーザーインターフェース) をツールベンダー各社が独自で開発し、自動化が設定しやすいように作成されています。

RPA ツールの多くが、エクセルのマクロのようにマウスやキーボードによる操作をその順序通り内部に記録し、フローを作成します。各処理の判断はルールベースで定義し、その実行の仕方は、独自の簡易言語等のスクリプトでカスタマイズできます。

どれくらい簡単に自動化の設定ができるかは、ツールによってこの UI の使い勝手に特色が出るわけですが、UI に統一された規格があるわけではなく、それぞれのツールによって設定の仕方は異なります。

一般的には想定されている業務がどのくらい限定されて設計されているかで、UI による自動化の設定の容易度が違ってきますが、容易さを追求するとその分カスタマイズがたいへんだったり、使用環境が限定されてしまう場合が多いです。

ともあれ、プログラミングなしで自動化できるという点では、この UI の部分が人気の秘密で、簡単に作業をロボット化できるテンプレート型が RPA ツールの大部分を占めております。



開発型の RPA

RPA が期待されている業務は多種多様です。

テンプレート型の RPA ツールでは、Microsoft .NET Framework が前提であったり、Windows 限定の稼働環境だったり、業務の種類まで限定されているものがあったりします。

UI で設定を簡単にしているため、どうしても複雑な処理をロボット化することは不得意だったりします。顧客特有の仕組みを取り込むために、RPA ツールに合わせて業務フローを修正することになれば、新しいシステムを入れることと同じで、運用を変更しないという RPA の良さを享受できない場合もあります。

それに対して、複雑な開発を簡単にするために汎用的プログラミング言語の API を使って、ソフトウェアロボットの作成を容易にしているのが開発型 RPA ツールです。たとえば、ROBOWARE は、Ruby や、JAVA、C#、PHP で API を使ってソフトウェアロボットの開発を可能にし、且つそのソフトウェアロボットを実行、管理するためのフレームワークです。

開発型の RPA であれば、新たにプログラム開発するより簡単に現行の業務運用を変更せず、複雑な顧客固有のバックオフィス業務を代行するソフトウェアロボットの作成が可能となります。

開発型ツールには、操作時のマウスの座標軸をプログラムコードに生成するジェネレーターなど、開発を楽にするためのユーティリティーがあります。

A P I API を使用してプログラミングによるロボット化



2つの型の比較

自動化したい業務が決まってい、予算的にも可能であればテンプレート型 RPA で対応できるものがあるかどうかを探ることが、即導入効果が得られ、検討時間の短縮につながります。但し、社内や委託先にプログラミングができる人がいれば、開発型の方が柔軟にカスタマイズもでき、低価格で満足度の高いソフトウェアロボットが制作できるかもしれません。

つまり、対象としたい業務内容や社内環境によって、最適な RPA ツールを選択することが重要です。導入実績や評判ももちろん大切ですが、自社の環境に合わないツールを選択した場合は、カスタマイズに時間がかかってコスト高になったり、運用変更を余儀なくされる場合もあり得るかもしれません。

また両方の型が、自動実行のためのスケジューラー等も搭載しているものが多いので、そうした運用も視野に入れて比較することが大切です。

RPA を成功させるためには、いろいろな角度から見てツールを比較検討する必要があります。

	テンプレート型	開発型
自動化設定方法	UIベース (ルールベース、マクロ、スクリプト等)	API プログラミング (コードジェネレーター等)
適用分野	特定業務 (ツールに依存)	広範囲
カスタマイズ	個別対応 (ツールに依存)	複雑な処理可能
エラー処理分岐	単一条件が多い (ツールに依存)	複雑な条件可能
業務フローの変更	おおむね難しい (ツールに依存)	対応可能
導入作業	個別対応 (ツール・環境に依存)	容易 (セルフレベル)
操作方法の習得	比較的簡単	プログラマーは簡単
運用・維持	比較的簡単 (ツールに依存)	簡単
初期コスト	サーバ集中管理型は比較的高価	比較的低価格
開発コスト	業務規模に依存	比較的低価格 (セルフ可能)

代表的な RPA ベンダー

ワールドワイドで有名な RPA ベンダーには、いろいろな特長があります。

Automation Anywhere (US CA)

RPA のソフトウェアベンダーではシェア No.1。Automation Anywhere Enterprise は、ソフトウェアロボットと共に構成され Windows 環境で稼働し、主な設定は、タスクエディタにより、自動化したい作業ステップを記録し、スクリプトを作成します。

Web サイトのデータ抽出やスケジュールされたファイル転送などの一般的なタスクを自動化する数十種類の事前構築タスクテンプレートが含まれています。光学式文字認識 (OCR) や Java との統合などの高度な機能を備えたオプションの統合パックを購入することで、幅広い外部アプリケーションと統合することができます。株式会社日立ソリューションズ等が、国内販売。

Blue Prism (UK London)

社名と同名製品の Blue Prism は、Microsoft .NET Framework 上に構築されます。様々なアプリケーションを、さまざまな方法 (ターミナルエミュレータ、シンクライアント、Web ブラウザなど) で提供されるあらゆるプラットフォーム (メインフレーム、Windows、WPF、Java、Web など) にて自動化します。比較的に価格は高めで、PCI-DSS、HIPAA、SOX などをサポートし、主に大企業向けの高いセキュリティが必要とされる領域に強く、サーバーでの中央管理方式のシステムを提供しています。RPA テクノロジーズ株式会社等が、国内販売。

UiPath (US NW : UiPath 株式会社)

UiPath は、ソフトウェア開発キット (SDK) の開発でルーマニアで設立され、現在米国を中心に活動しています。UiPath Studio は、コードの記述なしに、豊富なテンプレートライブラリーにより自動化の設定を記録できます。

UiPath Orchestrator は、ログやリソース管理、資産管理まで、全端末を幅広く管理します。UiPath Robot は、完璧な精度でプロセスを自動化します。



その他の RPA ベンダー

WorkFusion (US NY)

WorkFusion は、バックエンド作業をディープラーニングを使って改善することを目的に設立された AI のスタートアップです。WorkFusion RPA Express を企業向けに無料で提供するなど、戦略的な活動を行なっています。

Pegasystems (US MA : ペガジャパン)

Pegasystems は、自社の BPM や CRM を補完する目的で RPA の OPENSPAN 社を買収し、ロボット、分析、ケース管理を統合して提供します。日本をはじめ、アジア、欧州、北米の 35 箇所に拠点を構えています。

NICE (ナイスジャパン株式会社)

株式会社アイティフォー、ログイット株式会社等がパートナーとして販売。管理サーバーによる集中的な運用管理の方式で、ロックされた画面の背後で実行する機能やロボットの接続監視機能などがあります。

Redwood Software (US NC)

ERP プロフェッショナルサービスの経験があり、Redwood robots は、Oracle や SAP の ERP システムと連携して、物流や電子取引、財務などあらゆる分野での作業の効率化を実現します。日本では、リアルテックジャパン株式会社などが SAP 自動化等でパートナー販売しています。

Kofax (US CA : KOFAX Japan 株式会社)

OCR ソフトを富士ソフト株式会社等などが販売していますが、Kofax が買収した Kapow 社が Kofax Kapow という RPA 製品を持っており、RPA テクノロジーズ株式会社が、OEM 契約により Biz-Robo! として日本で大々的に展開し、株式会社日立システムズをはじめ、多くの企業が Biz-Robo! を提供しています。優れた UI を持ち、仮想マシンによるロボットの集中管理も可能です。



*Biz-Robo! は、ビズロボジャパン株式会社の登録商標です。

国内の RPA ベンダー

NTT アドバンステクノロジー株式会社 (川崎市)

純国産 RPA ツールの WinActor は、シナリオ作成機能がとても分かりやすく、PC 操作の繰り返し作業が簡単に自動化できます。他の大規模向け RPA ツールとは違い、サーバーによる中央管理の方式ではなく、個々の PC で手軽に導入できます。

シーティーシー・エスピー株式会社の販社をはじめ多くの企業で販売されています。

オルタフォース 株式会社 (新宿区)

IP ネットワーク環境上にあるソフトウェアロボットをコントロールする為のフレームワーク型の API を提供する Robowiser Framework は、IT 運用管理プロセスを自動化する Runbook Automation に始まり 20 年以上の実績が集大成されたソフトウェアです。

Robowiser Framework の OEM 提供を受け、株式会社イーセクターが開発型 RPA ツールとして ROBOWARE という名称で販売しています。

ソフトウェアロボットを開発できる言語は、Ruby、Java、PHP、C# で、高度なプログラミング知識なしでも、ソフトウェアロボットが作成でき、あらゆる規模のあらゆる業種の事務作業や、運用業務の自動化が可能になります。Windows と Linux で稼働します。



*WinActor は、エヌ・ティ・ティ・アドバンステクノロジー株式会社の登録商標、ROBOWARE は、株式会社イーセクターの登録商標です。

BPA と RPA

RPA と似た言葉に、BPA(Business Process Automation) があります。BPM (Business Process Management) が、人間、組織、アプリケーションなどのビジネスプロセスを設計、制御し監視することによって、ビジネスプロセスを管理し改善する活動を目指すのに対し、BPA は、ビジネスプロセス自動化のみならず、ビジネスワークフローの改善を行います。

年々導入が進んでいるディジジョン・オートメーションは、その BPA を具体的に補完するツールとしての役割もあります。

たとえば、株式会社アシストの Progress Corticon は、業務アプリケーションのディジジョン・モデルをルールフローに落とし込み、ルールの作成、分析、テストまでも可能です。プログラミングが不要なため、広義の RPA ツールの一つであるという見方もできます。

RPA がバックオフィスの事務・管理業務や情報システムなどの間接部門業務の自動化で注目を得ているのに対し、BPA はフロントオフィス業務のプロセス自動化が行なえます。但し、多くの RPA ツールは、フロントオフィス業務の自動化にも十分対応可能なため、例えばある RPA ベンダーの製品は、90% がバックオフィス、10% がフロントオフィスで活用されています。

本来、RPA のツールはバックオフィス系の業務自動化に改革をもたらすものとして脚光を浴びておりましたが、あまり BPA と区別する必要はなく、ERP や CRM といったシステムの自動化を補完するツールとしても活躍できます。RPA はデジタル労働者なので、ソフトウェアロボットとして自動化できる仕組みは業務の種類に関係なく、すべて RPA のカテゴリーに入れるべきかもしれません。



RPA の対象業務選定のポイント

RPA の導入を検討する場面において、最初に取り掛かるべき対象となる業務を選定する時のポイントは、現状業務のフローを変更しなくても RPA 化できる割と単純な定例業務を選定することです。

フローの変更をしたくない第一の理由は、導入効果の判定基準が複数になってしまい、RPA を導入したおかげで効率が上がったのか、業務フローを変更したから効果があったのか分からなくなるのを防ぐためです。

第二の理由は、導入期間を短縮するためです。業務フローの変更は、多くの人が関わり運用手順も変更になってしまうため導入前と導入後の影響分析にも時間がかかります。

第三の理由は、業務フローを変更してしまうと、基に戻せなくなるリスクが高まります。

こうした理由から、最初に RPA を導入する場合は、現在行なっている業務を誰かに引き継ぐ場合と同じで、オペレーションをソフトウェアロボットに引き継ぐつもりで業務選定することがポイントです。

RPA がうまく導入できれば、ソフトウェアロボットが PC 操作などの業務オペレーションをそのまま代行してくれます。

万が一、うまく稼働しない場合でも、業務フローが変わっていないので、従来通り人がオペレーションすることで業務が停止するという最悪の事態を回避できます。

このポイントに気を付けて業務を選定すれば、費用対効果を算定する場合も、誰かに引き継ぐ場合と同じように、人件費と比較するコスト、操作の速度、正確性を比較検証でき、RPA ツールを導入する効果があるかどうかの判定が非常にしやすくなります。



テンプレート型の選定ポイント

RPA の対象業務が決まれば、次は具体的な RPA ツールの検討になります。
今後 RPA の対象としたい業務が限定的であれば、RPA として人気の高いテンプレート型の RPA ツールが検討対象にできます。
処理が複雑でなければ、UI（ユーザインターフェース）が優れたテンプレート型 RPA ツールは設定が非常に簡単です。

① 単独の PC で動く業務かどうか？

多くの RPA ツールが、サーバでの中央管理方式ですが、中には単独 PC で動くことはできても、他の PC と連携できないツールもあります。
業務規模に合わせて、単独管理か集中管理方式かの確認が必要です。

② 標準の UI で作成できるか？

最初から、カスタイズすべき箇所が多いと、設定やテストに時間がかかりすぎてしまいます。
また、独自のスクリプトや、コーディング方法を覚えるのにもコストと時間が必要になってしまうため、標準搭載されている UI ですべて設定できれば簡単です。

③ コストメリットが出るか？

優秀な UI を搭載しているツールは、設定が簡単ですが、その分価格が高い傾向にあります。UI の使い勝手にこだわりすぎると、コスト高になる可能性があるため見極めが必要です。

④ 管理者が必要か？

サーバ中央管理方式の場合など、RPA ツールの管理者が必要な場合があります。管理者への負担やコストも考慮して検討すべきです。

⑤ 日本語対応、日本でのサポートが充実しているか？

海外製品の多いテンプレート型には、GUI が日本語に対応していないとか、日本でのサポート拠点が少人数であったりする場合もあります。導入後のサポート体制は重要です。

開発型を選んだ方がよい場合

テンプレート型の RPA ツールを検討し始めると、複雑なオペレーション業務などコストメリットが出づらい場合があります。今後 RPA の適用を広げていく計画の場合、むしろ最初から開発型を選択肢に選んだ方がよい場合もあります。
そこで、ROBOWARE を例に開発型ツールを選択した方がよいと思われるケースを取り上げてみます。

① ルールベースの条件では対応できない場合

エラー時の処理など、単純な条件分岐 (If ~then ~ else~ など) では対応できない処理の場合、テンプレート型の UI では自動化が設定しきれない場合があります。

② Linux 端末も混在している場合

テンプレート型 RPA ツールの稼働環境は、Windows 限定である場合が多いです。ROBOWARE の場合ですが、Linux が混在している環境でも大丈夫です。

③ 業務フロー、運用の変更が不可の業務

テンプレート型 RPA ツールを採用してその仕様に従い設定をする場合、業務フローや運用手順の変更を余儀なくされてしまう場合があります。

④ 複数個所から実行指示を与えたい場合

ROBOWARE の場合、多くのテンプレート型のサーバクライアント方式ではなく、どの PC でもサーバにもクライアントにも複数設定でき、相互にバックアップできます。

⑤ リモートで制御したい場合

テレワークのように、遠隔地からソフトウェアロボットを監視したり制御したい場合、ROBOWARE であれば、インターネットなど IP ネットワークがあれば可能です。多くのテンプレート型 RPA ツールは、セグメント越え、NAT 越えの操作命令等ができません。

⑥ 将来いろいろな業務に広げたい場合

テンプレート型では、得意とする業種や業務が限定されている場合が多いため、RPA の適用範囲を広げようとする、別のツールを購入する必要がある場合があります。開発型であれば、広範囲の複雑な業務までいくつも広げることが可能です。

AI を見据えた拡張性

RPA のハイライトは、AI です。主要な RPA ベンダーは、ディープラーニング等の機械学習や、自然言語処理を取り入れたコグニティブ技術をいろいろと取り入れています。自社開発しているベンダーもあれば、IBM Watson と提携した Blue Prism 社のようにコグニティブに強い会社と組んで提供しているところもあります。

いうまでもなく、RPA は繰り返し操作などルール化できる業務の自動化は得意です。しかしながら、非ルーティンタスクである直感的な判断や、創造性や問題解決能力が要求される業務は、AI 技術との連携が必要になります。そのためには、高度な技術と、高性能なマシン、高速なストレージが必要になるため、大手企業でやっと活用が始まったばかりの分野であり、今後ますます需要が増えてくるであろうと推測されます。

AI も日々進化を続けておりますが、企業の業務目的にあった AI のアルゴリズムを開発するには時間とコストがかかるため、従来のルーティンワークを RPA 化するケースとは分けて考える必要があります。

定例業務をそのままソフトウェアロボットへの代行させる場合は、割と短期間に適用が可能ですが、AI を搭載した RPA は、業務フローや運用手順の変更も入り、アルゴリズムのチューニングなど、検証テストにも時間が必要なため、適用目標期間を決めて特別なプロジェクトとして進めることが重要となります。つまり、AI 搭載の RPA 導入には、業務効率改善のためシステムを再構築するイメージが近いといえます。

AI もまだまだ分野が絞られ、汎用的な仕組みになりづらいところから、主要 RPA ベンダーが提供するコグニティブな自律型 AI を搭載した RPA ツールは、まだまだ適用範囲が限定的です。そのため、新規に業務改善を AI に求める場合は、開発型 RPA ツールを選択するのも有効な方法です。

コグニティブ技術の分野は、今後間違いなく発展していき、価格が安く汎用的になっていくことが期待できます。よって、初期段階ではまず、定例業務の操作代行としてのソフトウェアロボットにより、業務効率のアップや正確性の向上を進めることを最優先にすべきです。

RPA による業務改革は、個別の改善に留まらず、会社全体の業務の将来にわたっての長いスパンで考えて、導入計画をすることが重要です。



RPA 導入のためのポイント

RPA による業務改革を成功させるためには、プロジェクトとして進めることが大切です。

- ① 担当のアサインと役割・責任の明確化
- ② スコープ (適用範囲) 設定
- ③ スケジュール管理 (マイルストーン、実装時期)
- ④ コスト見積 (ツールライセンス、維持費)
- ⑤ 品質管理 (適合度、不具合時の対応)
- ⑥ リスクの洗い出しと対応策の管理
- ⑦ 導入判定基準の作成

RPA の特長は、高度なプログラミング技術がない人でも、業務操作の自動化を実現するソフトウェアロボットが作成できることです。そのために、テンプレート型のように優れた UI を搭載したツールがあるわけですが、たとえ開発型 RPA ツールを選択された場合でも、理想はプログラマがコーディングするのではなく、業務担当者が直接 API を利用しながらコーディングし、自らソフトウェアロボットを作成することです。

2020 年小学校でもプログラミングの授業が必須になる予定の時代において、実務担当の方がご自身でコーディングできれば、ソフトウェアロボットを作る立場となり、AI によって仕事を奪われる時代になっても優位に立てます。何よりも、実務を知っている方が作成される方が、リスクが少なく修正も楽で、開発効率も上がります。

RPA は、近い将来の AI による業務改革を見据え、まずは業務フローを変更せずソフトウェアロボットによる操作代行を目的にスモールスタートから始めることが、一番望ましいです。

ソフトウェアロボットの 作り方

2017年07月04日公開

<https://www.esector.co.jp/special/rpa/rpa3.html>



ソフトウェアロボットを作るには

ソフトウェアロボットがあれば、面倒な事務作業などの PC 操作が自動化できます。

通常ソフトウェアロボット制作するためには、かなり高度なプログラミング能力が要求されます。たとえば、マウスやキーボードの入力操作を一からプログラミングすることは、経験が少ないプログラマーや、Windows 等のパソコンの内部の仕組みを知らない人にとっては、とても難しく、時間もかかり、専門知識のあるプログラマーでなければ作成することさえ困難となります。

そのため、多くの RPA のツールベンダーは、メモリやストレージがいろいろ選択できる BTO パソコンのようにシステム化して、GUI で比較的簡単に設定できるソフトウェアロボットを業務に合わせ、いろいろな種類のパターンで作成しました。これにより希望するソフトウェアロボットの形に近いものを選んでカスタマイズできれば、簡単にソフトウェアロボットが作成できます。

しかし、企業や組織の業務は多種多様で、すでにある形に無理に合わせようとすると、希望するソフトウェアロボットが作成できない場合も多いです。

自作パソコンをイメージしてみてください。まず PC のフレームを用意して、メモリやディスクやファンなど、すでに完成された部品を利用するでしょう。気に入った部品を揃えて組み立てれば、そんなに高度なパソコンの知識がなくても、思い通りのパソコンができるはずです。

ソフトウェアロボットについても、同様にフレームワーク化された API を搭載した ROBOWARE を使用すれば、高度なプログラミング知識がなくても、比較的容易に作成することができます。



目的に合ったソフトウェアロボットを作成するために、Ruby や PHP などの汎用プログラミング言語を使用して制作できる ROBOWARE での作り方と実行方法を解説します。

ROBOWARE --- ソフトウェアフレームワーク

ROBOWARE は、ソフトウェアロボットを開発、実行するためのソフトウェアフレームワークです。

ソフトウェアフレームワークとは、ソフトウェアロボットの開発・運用を行う際に、その基礎となるルール・構造・アイデア・思想などの集合で、ROBOWARE にはソフトウェアロボットを作成するために多くの API が含まれています。

ROBOWARE のソフトウェアフレームワークにより、ソフトウェアロボットの骨格が用意され、開発も容易で、実行時にソフトウェア同士の連携が簡単に取れます。

プログラミング言語としては、Ruby、Java、PHP、C# にて作成されるスクリプトに対応したソフトウェアロボット専用のソフトウェアフレームワークです。



フレームワークのメリット

ソフトウェアロボットを作るために必要な枠組みはすでに出来ているため開発コストが下がり、ある程度の品質が保証されます。また、設計、構成などがフレームワークとして用意されており、機能が明確に分離されているため、保守性が上がり、またコードが読みやすくなる、というメリットもあります。

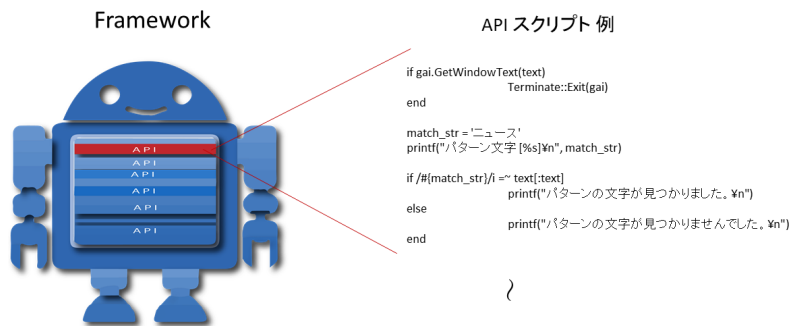
ROBOWARE --- API

ROBOWARE は、ソフトウェアロボットを開発し、実行・運用するための多くの API (Application Programming Interface) を実装しております。

API とは、ソフトウェアが、他のソフトウェアとお互いやり取りをするために機能を共有出来るようにするためのインターフェースのことです。
たとえば、異なるアプリケーションを連携させるためには、通常内部のレコードフォーマットなどの仕様に合わせてアクセスできるようにしたインターフェースが必要なため、目的に応じた API を用意します。

ROBOWARE の場合の API とは、ROBOWARE に搭載されたソフトウェアフレームワークに対して、ソフトウェアロボットを開発、実行するためにスクリプトにより指示できるように実装されたインタフェースです。
この API に引数として、指示にひつような情報を与えることによって、ソフトウェアロボットとして稼働するための動きが指示できます。

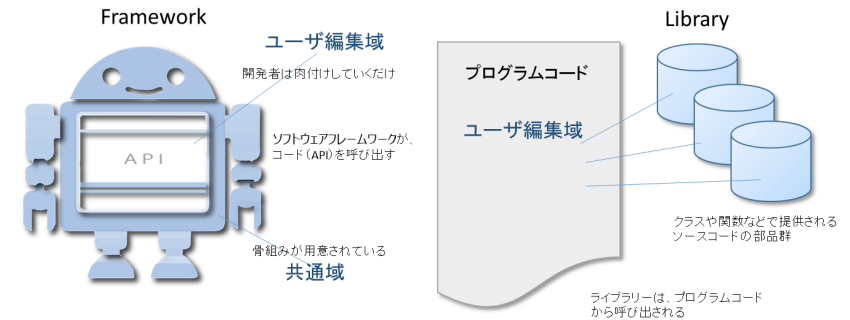
API を利用することで、どのような文字列をコピーしたり、クリックするかなどを複雑なプログラムコードを記述することなく、引数によって簡単に指示できます。
ROBOWARE の API の引数は、基本として ハッシュ型を採用しております。ハッシュは、連想配列とも呼ばれ、文字列を キー とする配列で、1 つの API に対し、いろいろな指示が可能です。



ソフトウェアフレームワークとライブラリの違い

ROBOWARE は、ソフトウェアロボットが制作できるソフトウェアフレームワークです。

ソフトウェアフレームワークは、定義された API を持ち、具体的な実装を再利用可能な形で持っている点でライブラリとよく似ています。
しかし、ライブラリではプログラムの制御に関する主導権がプログラムコードにあるのに対し、ソフトウェアフレームワークでは、フレームワークがコード (API) の制御を行ないます。



プログラムのライブラリは、通常共通で使用できるプログラムの部品をひとまとまりのファイルにしたものです。
ソフトウェアロボットの場合、そのライブラリに共通部品を置いて使用したとしても、メインとなるプログラムコードを記述しなければならず、通常のプログラム開発をする工程と同じとなります。

一方、ソフトウェアフレームワークである ROBOWARE は、既にソフトウェアロボットの枠組みが構成されているため、API に引数を渡すスクリプトをコーディングするだけで、ロボットに指示が与えられます。データの選択条件や、エラー時のハンドリングなどを詳しく指定する必要がある場合でも、通常のプログラム言語によるコーディング追加によって、ロボットに対し更に詳しい指示が可能になります。
また、ソフトウェアロボットが動くためには、別の PC からの指令や、リモートから制御する仕組みが必要な場合があります。ROBOWARE であれば、単独のプログラムと比較して、実行・管理が出来る仕組みも予めフレームワークに備わっているという利点もあります。

ROBOWARE の導入

ソフトウェアロボット制作のために、ROBOWARE を、操作を代行する PC に導入します。
稼働環境：Windows および Linux
インストーラーの起動で簡単に導入が出来ます。
(使用中のウイルス・スパイウェア対策のソフトに対象外の設定をする必要があります。)

- RBF API : ROBOWARE が提供する各種 API
- RBF サーバ : RBF API の命令に基づき処理を実行、または通信リレーを行うサーバソフトウェア
- サンプルスクリプト集 : API のコーディング例

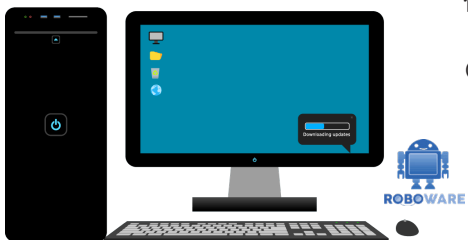
ROBOWARE の導入として、RBF ソフトウェアをインストールすると、上記、3つが使用可能になります。

導入後、マニュアルも導入フォルダ内に格納されます。
各 API、サンプルスクリプト集もプログラミング言語ごとに 4 種類用意され、Ruby、Java、PHP、C# に分かれてそれぞれのリファレンスマニュアルがあります。
RBF サーバのインストール後、RBF アプリケーションを起動するランチャーのショートカットが設定されます。

次に、ROBOWARE で開発したプログラムを管理するための Job Manager をインストールします。

Job Manager を導入すると、ROBOWARE で作成されたソフトウェアロボットの実行や管理ができます。

(導入後、ライセンスの登録が必要です。)



サンプルスクリプトを選ぶ

API で使用するスクリプト用の言語を Ruby、Java、PHP、C# より選択します。

Ruby、Java、PHP、C# の各言語ごとに用意されている 50 種類以上のサンプルスクリプトファイルより、参考になりそうなサンプルコーディングを選びます。

各言語のマニュアルを基に、API を使い、ROBOWARE にどのようにオペレーションするかを、プログラミング言語によって指示します。

このサンプルを参考に、言語の種類ごとユーザが使い慣れたエディタを使用してコーディングします。

```
# encoding: sjis
#-*- coding:
=====
RBF_BaseAPI
=====
【サンプル】
Windows OS
再起動時は
【サンプル】
RBF_BaseA
RBF_Base
MagicPack
【サンプル】
RBF_BaseAPI for Windows Ruby Sample
=====
【事前条件】
1.再起動さ
している必
2.再起動さ
行されてい
【サンプル】
1. Shutdown
2. Windows
=====
【事前条件】
1. Window
2. デスクト
require 'RBF
=====
【GUI制御】
Windows
制御したい
ボタンで
何かしら
更されてい
=====
require 'RBF
=====
【サンプル】
1. GeWind
require 'RBF_BaseAPI'
=====
end
=====
【エラー】
=====
module Terminate
module function
def Exi(gai)
#最後に失敗したエラーコードを表示
print("Error Code[#{@g}]")
gai.LastErrorCode
#最後に失敗したエラーコードに対するエラーメッセージを表示
print("Msg[#{@g}]")
gai.LastErrorMsg
gai.Close
end
end
end
=====
# RoboticAPIクラスのインスタンスを生成
init_def = {
# RBFサーバとAPIとの通信タイムアウト(秒)(Default: 180)
# timeoutには、RBFサーバとAPIとの通信で、通信送受信
# timeout値を超過した場合、全ての通信経路(リレー)
# されます。省略した場合、デフォルト値となり
:timeout => 60
}
gai = RBF_Base::RoboticAPI.new(init_def)
```

RBF API プログラム開発 リレー通信

ROBOWARE の開発言語基本記述構成

目的のRBFホストへコネクト
(RBFリレーソケットオープン)

命令処理
(操作、リソース取得、判断)

RBFソケットクローズ

(Javaの例)

```
int timeout = 30;
gai = RBF_Base::RoboticAPI.new
RBF_BaseAPI.Open connect = new RBF_BaseAPI.Open();
}

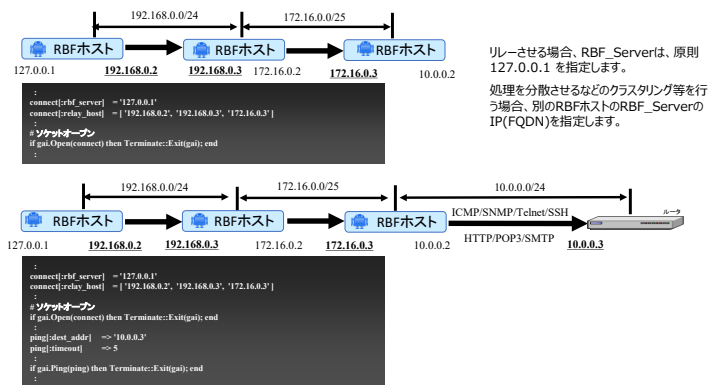
RBF_BaseAPI.GetPID pids = new RBF_BaseAPI.GetPID();
pids.proc_name = "RBF_Server.exe";

if(gai.GetPID(pids)){
    Exit(gai);
}
System.out.printf("PID [%d] %n", pids.pid[0]);
}

gai.Close()
```

* Ruby, Java, PHP, C#が使用可能

配列変数にリレーさせる経路順に IP(FQDN) 設定します。



ROBOWARE は、処理したい目的の RBF ホスト（サーバ）が別のセグメントにある環境など、ROBOWARE に作成した API からの命令を、通信途中にある RBF ホストでリレー（中継）させることができます。

リレーする RBF ホストは、複数中継させることができます。

API の種類

ROBOWARE は、Windows または Linux で動作可能な約 80 の API を提供します。

ROBOWARE の API に、プロセス名や座標などの引数を与えるだけで簡単にスクリプトが作成できます。

ROBOWARE APIの例	
GetActiveProcessNum()	実行されているプロセス数を取得
GetCPUUsageRate ()	指定のPIDまたはOS全体のCPU使用率を取得
AddRemovePrograms()	インストールされたソフト一覧の取得/アンインストール
ExecProcess ()	プロセスや実行ファイルを起動
FileTransfer ()	ファイルを転送
SendMail()	メールを送信
WindowCapture()	ウィンドウをキャプチャーする
KeyboardTyping()	指定のウィンドウハンドルにキー入力
TermPrint()	Telnet/SSHサーバにコマンドを送信する ...etc

スクリプトを実行する RBF ホストの OS が Windows の場合に、端末でソフトウェアロボットが稼働する RBF ホストの環境が Linux であっても制御可能です。

同様に、スクリプト実行 RBF ホストの OS が Linux で、端末 RBF ホストが Windows でも可能ですが、端末 RBF ホストの OS で指定できる引数が異なる場合があります。

ROBOWARE には、多数のカテゴリーの API があり、各プログラミング言語のリファレンスマニュアルには、該当 API がどのサンプルスクリプトに例文があるのか記載されています。

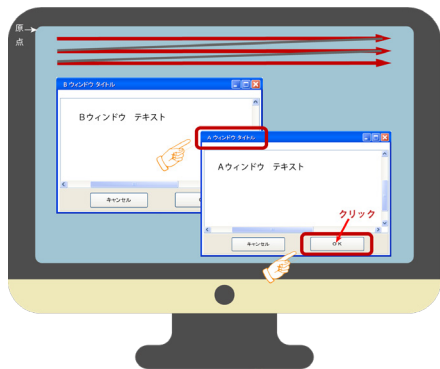
- TCP 接続開始、終了、エラーに関する API Method
- OS またはプロセスのリソース取得設定に関する API Method
- OS またはプロセス制御に関する API Method
- 通信に関する API Method
- メールに関する API Method
- Windows UI 制御に関する API Method
- . . . 他

ROBOWARE の画面表示認識

モニターに表示されているウィンドウのテキスト文字を RBF API の引数に指定して、目的の操作 (クリック・キー入力等) を行います。

遠隔で制御したいロボット (RBF ホスト) のモニターの解像度等に関係なく開発することができます。

- ① ウィンドウに表示される「親ウィンドウのみ」または「親ウィンドウと子ウィンドウ」の2つのテキスト文字を引数に指定して検索します。
- ② モニターに表示されているテキスト文字を検索する場合、ワイルドカード検索もできます。
 - ・Windows OS が異なる場合で「サーバー」と「サーバ」と表示が異なった場合でも「サーバ*」と指定することで検索することができます。
- ③ モニターに表示されている文字で同じ文字が複数あった場合は、検索して見つかった順番の番号、または Z オーダーを引数に指定します。
- ④ モニターに表示されていないテキスト文字の場合 (モニター枠からはみ出している場合など) は、検索することができません。
 - ・但し、ウィンドウがモニター外にはみ出していると判断した場合、自動的にモニター内にウィンドウが移動します。



ROBOWARE アナライザー

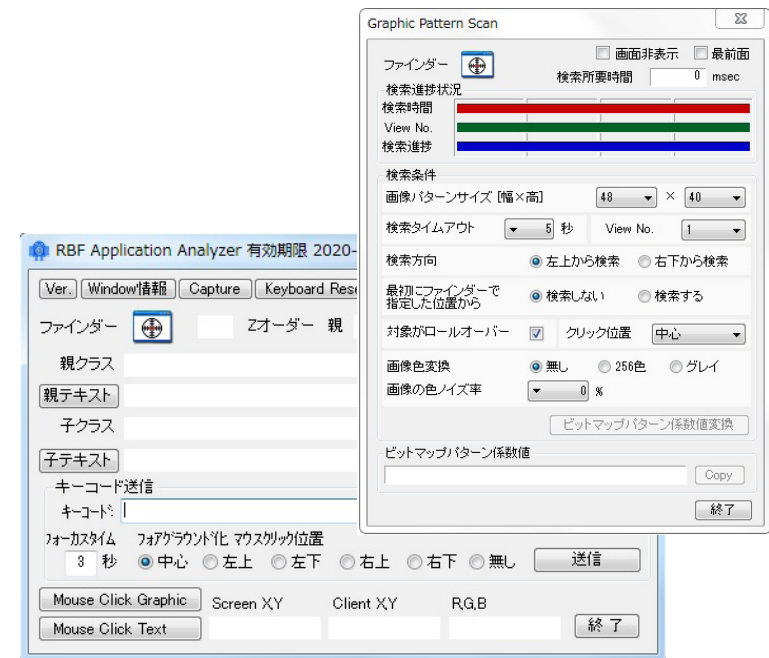
マウス・キーボード操作の開発は、RBF アナライザーを使用します。

自動化したいアプリケーションの画面を基にマウスやキーボード操作について、RBF アナライザーを使用して RBF API のメソッドの引数に渡す情報を取得できます。

これにより、画面上の操作を簡単にスクリプトコードにジェネレートできます。

典型的な RPA ツールであれば、ツール独自の GUI によって、スクリーン上の画像の座標を自動認識し、連続した操作を設定できるものが多いですが、ROBOWARE は、いろいろな条件で操作をハンドリングできるように、API の引数の情報をアナライザーによって取得し、その後、作成者自身でスクリプトに反映する方法となります。つまりプログラミングを支援するユーティリティとなります。

ROBOWARE は、スクリーン上のテキストとグラフィック両方のパターンを認識できます。



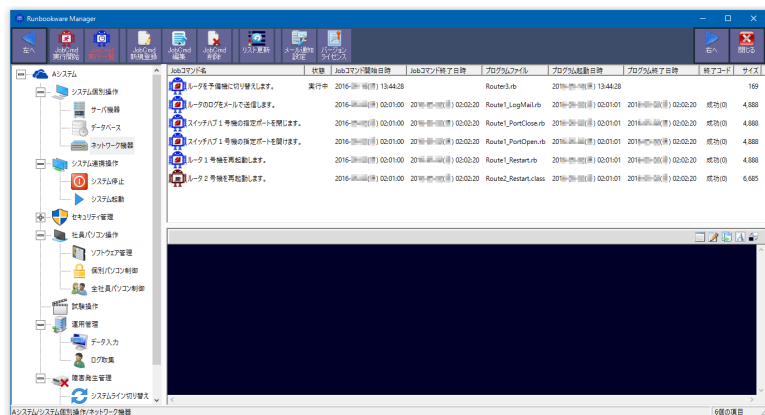
ROBOWARE プログラム管理

ROBOWARE – Job Manager

ROBOWARE で開発したソフトウェアロボットを実行・管理

RBF API で開発したプログラムを Job コマンドとして、管理するソフトウェアです。

Job Manager の画面は、タッチパネルモニターの場合に、Job コマンドを実行するまでの操作を指でタップし易いように設計されています。



- Job の実行スケジュール機能
- Job の開始と終了結果のメール通知機能 など

よく使用するランチャーボタンを一覧表示し、Job コマンドを用途別にフォルダのツリー形式で管理でき、Job コマンドの一覧が出るエリアと、詳細を表示するエリアに分かれて表示されます。

分割されたエリアのサイズ変更や、フォルダのアイコンの変更も可能です。

Job コマンドの新規登録

作成したソフトウェアロボットを実行するために Job Manager で Job コマンドを作成します。

Job Manager の Job コマンドツリービューに表示されたフォルダ毎に、Job コマンド登録情報を作成できます。

実行する Job について、プログラムファイルのパスや、プログラムを実行する時の引数、起動時に引数を入力させるかどうかなどを指定できます。

また、Job コマンドリストビューのから選択で、Job コマンド実行方法の設定ができます。



ソフトウェアロボットの実行方法

ROBOWAREには、スケジューラ機能もあり、作成したソフトウェアロボット毎に Job Manger で予め実行形態を登録できます。

- ・即時実行
- ・時間サイクルで実行
- ・指定日時で実行
- ・週サイクルで実行

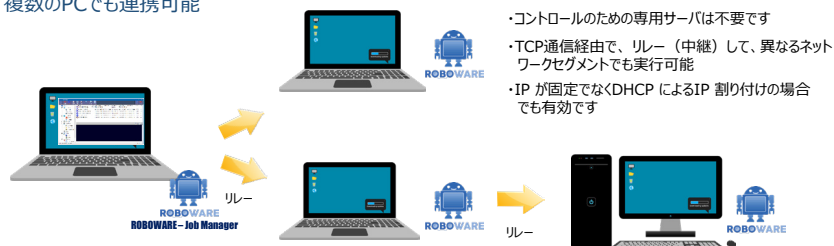


バッチあるいはユーティリティコマンドからも起動停止が可能なので、他のスケジューラへの組み込みや、アプリケーション連携が簡単に行えます。

ROBOWAREは、他のROBOWAREが導入済みのPCやサーバに対し、実行指令が可能です。これによって、IPネットワークで接続されていれば、テレワークに対応して、遠隔地の自宅やサテライトオフィスからでも、制御することが可能になります。

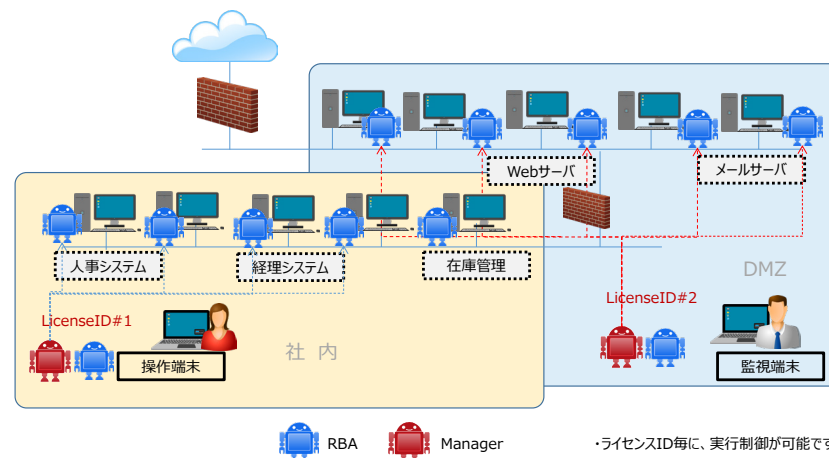
中央管理のための専用サーバは必要なく、ROBOWARE同士で、異なるネットワークセグメントにおいてもリレー（中継）して実行することが可能で、相互のバックアップ（冗長化）になることもできます。

複数のPCでも連携可能



ROBOWARE 構成例

操作端末からの操作は社内のみ、監視端末からはDMZのみの端末操作を行う場合の例



ROBOWAREは、ライセンスIDでお互いの通信を制限できます。同じ社内の異なるシステムにROBOWAREが共存する場合であっても、外部のROBOWAREからそのシステムを制御されない作りになっております。

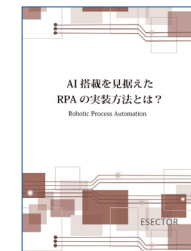
典型的な例では、実行したいPC上にROBOWAREを導入し、それぞれ作成したソフトウェアロボットが動く環境にします。実行指示や、実行結果の管理を行うPCあるいはサーバには、ライセンスID毎にJob Manegerを導入して制御します。

ROBOWAREでソフトウェアロボットを作成すれば、様々な業務形態に対応して比較的簡単にオペレーションの自動化が実現できます。

AI 搭載を見据えた RPA の実装方法とは？

2017年09月27日公開

<https://www.esector.co.jp/special/rpa/rpa4.html>



働き方改革の要は RPA

働き方改革が多くのメディアに取り上げられ、話題になっています。

近年、洗濯は乾燥まで全自動となり、お掃除ロボットや自動食器洗い機などのおかげで主婦の家事労働に余裕ができ、ジムに通ったり趣味に時間を費やすことが可能となりました。

一方、職場はどうでしょう？

コンピュータで行うべき難しい処理は、マシン室のサーバに集められスケジューラなどいろいろな効率化のツールを使い、その多くが自動化されてきました。それなのに、オフィスで働く人のほとんどは、未だにパソコンに向かいキーボードを打ちマウスを動かしています。

このような状況で、どのようにしたら仕事に余裕が生まれるのでしょうか？

それを解決するには、ロボットに仕事を任せてしまうのが一番です。

たとえばテレワークの在宅勤務を例にした場合、分かり易いのがソフトウェアロボットというデジタル労働者に仕事を任せて、自分は家からその労働者を見守り指導するという方法です。もちろんセキュアなシステムであることが前提ですが、在宅勤務を可能にするためには、まず会社で行なっている仕事を自動化することが必須です。会社での PC 操作を不要にすればじめて家での仕事が可能になります。それを現実的に実現してくれるのが RPA です。

RPA(Robotic Process Automation) は、デジタル労働者のソフトウェアロボットを使って PC 操作も含め業務を完全に自動化することを目指します。

産業用ロボットが活躍している工場では、実際に作業するのはロボットで、そこで働く人はそのロボットの監視やチェックが主な仕事となっています。一方、オフィス業務のロボット化では、ロボット自身のハードウェアのメンテナンスがいらないため、ソフトウェアロボットに作業を任せても、その監視さえも担当者は自宅からリモートで実施でき、且つ実行指示を与えることができるようになります。だからこそテレワークが実現可能となります。

働き方改革 ———— ロボットが活躍



RPA を社内で進化させる

RPA はテレビ番組などで取り上げられたためホワイトカラーの雑用の自動化で脚光を浴びました。しかし、RPA の可能性は、それだけで終わらせては勿体ないです。

もちろん RPA を初めて導入する場合は、人が行っていたオフィス業務の単純作業を自動化するだけでも充分効果があります。しかし、RPA は構築次第で複雑な業務も対応が可能なのに、多くの人たちが RPA の可能性を理解できておらず、パターン化できる作業しか自動化できないと思ってしまうようです。

この仕事は自分にはできないとか、経験がないと他の人には無理だと思われる業務は多いです。この固定概念があると特に経験を積んだベテラン社員は、いつまでもその仕事に縛られてしまうことになりかねません。

RPA の導入で、ソフトウェアロボットという後輩に、どんどん仕事を引き継ぐイメージをしてみてください。最初は、単純作業から始めて、いろいろ経験をさせることで難しい仕事も引き継ぐことができます。この時、なるべくソフトウェアロボットが仕事を自動化しやすいようにとわざわざ業務フローを変更しないようにして下さい。フローを変更すると、いざという時に、あなたが代わって手続きする場合に手順が分からなくなってしまいます。

無理だと思っても、引き継ぐことをイメージすると自分がしていた作業をパターン化して整理することが可能になります。経験による判断が必要な場合、引継ぎ書にすべてのパターンを洗い出してケース毎に処理する手順を明確化できれば、ソフトウェアロボットにプログラムすることで指示がしやすくなります。

プログラムで作成するのは複雑で難しい場合、今まで手計算に頼っていた部分をパターン化することでエクセルのマクロ等で作成できてしまう場合があります。それを利用すれば、ソフトウェアロボットはオペレーションに徹して、画面の数字をエクセルに渡し、エクセルのマクロの計算結果を他のアプリ画面に自動入力させるなどといった形でとても効率化できます。もっと複雑な場合、交通費計算や経費精算など、特定分野の効率化ツールと RPA ツールを連携させてしまえば、今まであなたの判断が必要だった業務も少しずつソフトウェアロボットへ引き継ぎ、RPA を進化させながら業務を効率化させることができます。



RPA で期待される AI

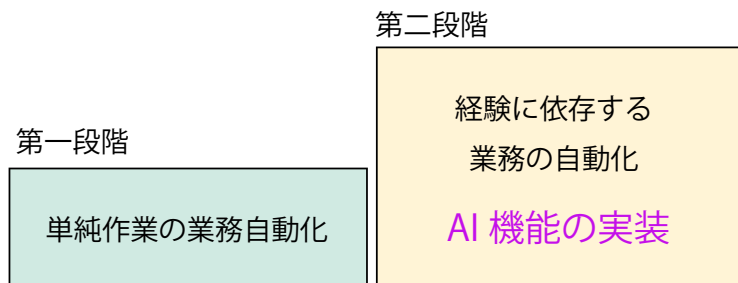
第4次産業革命の核となっているひとつに AI（人工知能）がありますが、生活の中にも AI 搭載のロボット家電が一般的に普及し始め、RPA のソフトウェアロボットにも AI を搭載して実装する期待が高まってきています。

初期の RPA の目的は、キーボードやマウスの操作を中心とした定例業務の自動化で十分でした。

RPA ツールを使えば、エクセルでマクロ登録をするレベルの技術力で、単純作業の自動化が簡単に行なえます。定例業務のルーチンワークが自動化できれば、速さも正確性も増し、RPA の効果が実感できました。するともっと他の複雑な業務にも拡大して自動化を推進していきたい、という RPA への期待が高まります。

誰でも操作ができそうなエクセルやワードなどの業務であれば問題ないのですが、人の判断にゆだねる操作や、メール文書のようなパターン化しづらいデータを処理する複雑な業務は、残念ながらその構造上 RPA ツールの多くが不得意です。しかしながら、ロボットは賢いというイメージにより、人の経験や知識に依存する業務などに対しても自動化したいという要望も多くなってきています。というのも、働き方改革を望んでいるのに、ベテラン社員はソフトウェアロボットの助けを借りられないのであれば、いつまで経っても在宅勤務が夢で終わってしまうからです。

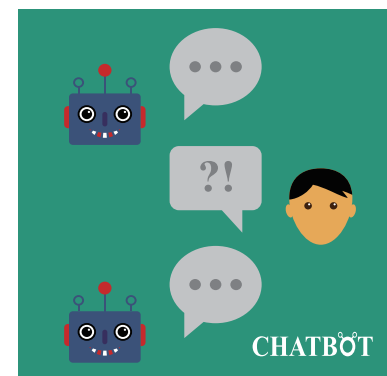
そこで、RPA の進化の第二段階として人工知能である AI の搭載が目まぐるしく注目されています。メディア等で AI の先進技術が取り上げられると、コンピュータの知能の力を借りればほとんどの仕事が代替できるはずだという期待が、RPA の分野にも浸透してきています。



業務に必要な AI とは？

AI を必要としている目的が、あたかも人が判断したような動きをすることであった場合、業務によっては、そんなにコストをかけなくても高度な AI を搭載しているかのようにシステムを自動化することが可能になります。

その代表例が、チャットボットです。「人工無脳」（あるいは人工無能）と呼ばれ、テキストや音声を通じて、会話を自動的に行うプログラムですが、実体は会話の中のキーワードを拾って、内部のデータベースとマッチングすることにより、シナリオに従って選択的に会話したり、会話ログを利用して文脈に適切と思われる応答を返しているだけです。



それでも、コールセンターやヘルプデスクなどでは大活躍しており、その人間らしさには驚かせられます。チャットボットを多くの人は、自律型の本格的な AI が搭載されていると信じて疑わないほど、バックグラウンドで動く仕組みが巧妙に人間らしさを追求されているものが多いです。

そこまでしなくても、簡単な業務では人が行なっているかのような動きを与えることができる場合もあります。

たとえば、お客様から来る資料請求のメールに対して、商品の種類が多数存在している場合でも、RPA ツールで送るべき資料を自動的に判断できる様にプログラムしてしまえば、あたかも担当者が手続きしたかのように迅速に資料を送ることができます。その際、見積依頼を定型フォーマットで入力できるように Web 公開しておけば、複雑な見積書の自動返信にも応用できます。

もちろん、他社に先駆けて今後の業務拡大を行なうためには、自律型 AI の技術が必要な分野もありますが、コストを考えた場合、人らしさを考慮すれば本格的な AI を搭載しなくても、プログラミング次第で対応できる業務は結構多いというのが実情です。

業務に必要な AI は、意思決定をしてくれる仕組み

AI って何ですか？

人工知能つまり AI は、コンピュータを使って創造された知能ということになりますが、そもそも生物でないコンピュータに知能が備わるのでしょうか？

知能を単に物事を理解したり判断したりする力と定義してしまえば、その理解力や、判断レベルで AI もいろいろな種類に分けられることになります。

幼児であっても知能が備わっているといわれるように、知能と同様コンピュータにどこまで判断させる機能を備えさせるかによって、コンピュータの AI のレベルも違ってきます。

本来であれば、コンピュータ自身が考え行動するという能力が AI に求められていますが、状況によって知能という言葉の解釈が変わるため、世間では単純にルールに従って出された受け答えでも、人間らしく振舞っているように見えるコンピュータの動作はすべて AI という言葉で代替されているようです。

AI の歴史は古く、1956 年の「ダートマス会議」で初めて人工知能 (artificial intelligence) という言葉が使われたといわれています。この頃の第一次 AI ブームでは、人工知能は「推論と探索」を中心とした手法が盛んで、それはルールやゴールが決まっているチェスや迷路などを解くためのものでした。但し、ヒトの脳を模したニューロンという理論もすでにこの頃から研究が始まっていました。

その後、1980 年以降の第二次 AI ブームでは、世界中の企業で「エキスパートシステム」が採用されるようになり、日本でも第五世代コンピュータプロジェクトが開始されました。

エキスパートシステムは、特定の分野で知識ベースを使って推論を行なうルールベースのプログラムで、人間の専門家のように意思決定ができるようにしたシステムです。

そして、2010 年以降現在も盛り上がっているニューラルネットワークを使った「ディープラーニング」によって、第三次のブームがやって来ています。

有名な囲碁専用の AI 「AlphaGo」が、人間のプロ棋士に勝利して依頼、Google の翻訳や、自動運転技術やトレーディングなど、深層学習を実用化したシステムが多く登場してきています。



弱い AI と強い AI

人工知能を弱い AI、強い AI という言い方で分類することがあります。

弱い AI というのは、ルールベースに始まり特化した分野のエキスパートシステムまで、あたかも優秀な人間を模して処理されるのに対して、強い AI は自ら学習し、自律型で汎用性がある AI を指しているようです。

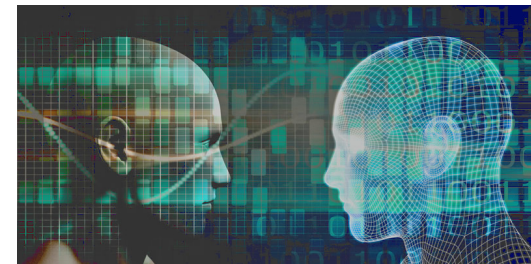
近年のディープラーニングのブームに乗って、多くの大企業や組織によって汎用人工知能 (AGI: artificial general intelligence) の研究が積極的に進められています。

AGI は、人間レベルの知能の実現を目指しているため、スーパーコンピュータを使用して神経細胞とシナプスからなる大脳皮質をシミュレートするなど、人間の脳についての研究も進められております。

従来のルールベースの推論では難しかった音声認識、画像認識等でかなり実用化されて来てはおりますが、人間の脳に迫る汎用的な知能というものは、判定も難しくまだまだ実現できる分野が限定的であるようです。

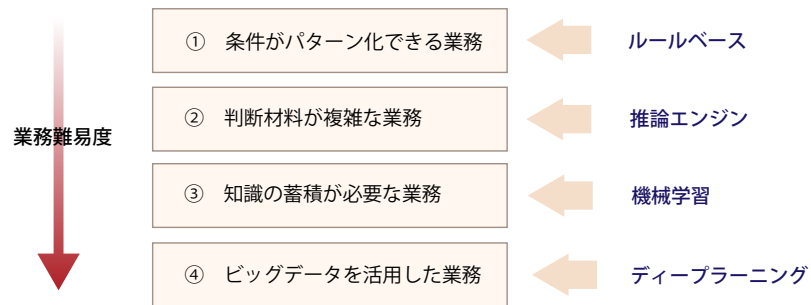
意識を持ったコンピュータの能力を強い AI とするならば、本来人工知能という AI は自分の意志を持って自律できる人工知能を目指して開発されて来ていたので、本当の意味での AI の姿は、レイ・カーツワイルの予言で有名になった「シンギュラリティ」(技術的特異点) に到達したときに完成するのかもしれませんが、そんな時が到来すれば、人類はコンピュータと融合し、AGI を搭載したコンピュータは自分達自身で、自分の分身をいくつも量産可能になるかもしれないという恐怖もあります。

強い AI はまだまだ発展途上ですが、現実的には第 2 次 AI ブームで多くの企業が採用したエキスパートシステムのように、弱い AI が多くの企業で活躍し始めています。ルールベースのパターンマッチングを利用した推論や探索に基づく弱い AI であっても、普通の業務であれば、人間の処理速度や処理量をはるかに超えてミスなく処理できるため、充分役立っています。



業務の分類と AI 技術

RPA ツールは、通常頭脳にあたる AI 機能を持っていません。中には積極的に取り込んでいるツールもありますが、全ての要件をカバーできる AI はないため、業務によって柔軟に組み合わせることができる RPA ツールとその業務に合った AI 技法を選ぶことが大切です。



単純な作業でパターン化できる業務は、ルールベースによって自動化できます。多くの RPA ツールは、数種の条件であればルールベースを組み込んで、オペレーションを自動化する設定が可能となっています。

複雑なパターンが複数組み合わせたり、場合によって適用すべき条件を変更しなければいけないような複雑な業務は、推論エンジンを使用することが効果的です。過去のデータを元にいろいろな事象を総合して答えを導き出す場合など、答えが 1 対 1 の比較だけでは断定できないような場合に有効です。

知識の蓄積が必要な業務は、日々の経験の積み重ねによる判断が必要な業務ですが、これは最新データも常にリアルタイムで追加しながら、判断の対象とすべき内容のため、機械学習の技法が必要です。

ビッグデータを活用した業務は、機械学習の中でもディープラーニングが向いています。とてつもなく大量のデータを学習させるためには、処理するコンピュータ自らが自律して処理できないと、そもそもデータ量が多すぎて答えを導き出すためのデータを与える人間の処理能力が追いついていきません。

具体的にそれぞれの技法を見ていきましょう。

ルールベースの AI

ルールベースの AI とはどのようなものなのでしょうか？

判断を求められる処理に対して、その条件に対して行うべき処理をあらかじめ登録しておくことにより、コンピュータに答えさせる方法です。

If ~ then ~ で表せるような、ある条件に対して、どのようなアクションを行うかという、プログラミングであれば最も基本となる If 文形式のものが当てはまります。

プログラムの中で、If 文や、Switch 文を使用してそれぞれの場合に合った処理方法を記述しておけば、状況による判断をあらかじめ指定しておくことが可能になります。

判断内容によってはコーディングする行数は多くなりますが、プログラム初心者でも簡単に、コンピュータを人間らしく見せることができるベーシックな手法です。

これを応用して、else をつければ条件に該当しない場合のアクションを選ぶことができ、If で尋ねる条件をテーブル化すれば、量が多い場合の条件にも対応できます。

初期の頃のロールプレイングゲームは、このパターンでの受け答えが多かったのでわかりやすいでしょう。

ルールの量が、エクセルでテーブル化できるレベルであると、エクセルのマクロや VBA プログラムで簡単に作れます。

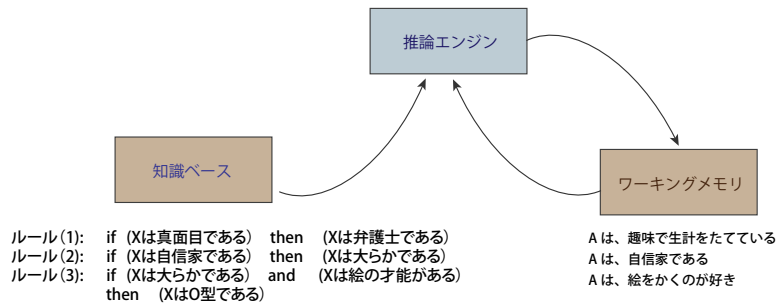
セキュリティ上エクセルのマクロを禁止されている場合でも、If 関数で記述するだけで、If ~ then ~ else ~ の処理を簡単に登録できます。

RPA でこの処理を自動化するためには、RPA ツールにはキーボードやマウスの操作のオペレーションのみを設定し、ルールベースのコーディングは RPA ツールに記述するのではなく、エクセルのような表計算ソフトや、専用のルールベースを処理できるツールに任せれば、かなり多くの判断が必要なデータの処理であっても楽に設定でき、高速かつ正確な処理が実現できます。

<pre> ルール (R1): if (条件式:S1) then (実行文:EX1) </pre>	<pre> Switch(条件式) { case 数値: (実行文:EX-3-1); break; case 数値: (実行文:EX-3-2); break; case 数値: (実行文:EX-3-3); break; } </pre>
<pre> ルール (R2): if (条件式:S2) then (実行文:EX2-1) else (実行文:EX2-2) </pre>	

推論エンジンと知識ベース

エキスパートシステムで有名な推論エンジンですが、その仕組みはルールベースを発展させ、知識ベースの情報を元にワーキングメモリを使って推論することを可能にします。代表的な「PS (Production System)」の例では、知識ベースに事前に様々なルールが登録されており、ワーキングメモリつまり作業領域を独立して持っていて、そこに事実や特徴のデータを格納しておきます。そして、知識ベースに入っているルール郡と、ワーキングメモリに蓄積された内容を推論エンジンが照合して、回答を導き出します。下記の A は画家だと推論できます。



推論エンジンは、知識ベースにある条件を、ワーキングメモリ内の内容とマッチングさせ、それに当てはまった規則はすべて実行すべき候補とするため、その中で実際の規則を実行すべきか競合解消を行ないます。その場合、最近実行した規則にマッチしたデータが高い優先順位になったりします。データ格納部は、推論エンジンにより選択され実行されることによって更新されるため、次のサイクルでマッチする規則の集合も変化します。このように情報が雑多なため、その処理にはかなり負荷がかかり、いろいろな高速化・効率化する手法があります。ルールベースでは、論理的なある意味の断定に近い回答を得られていましたが、推論エンジンの発達で抽象的な内容についても判断ができるようになってきました。状況に応じて物事の判断に曖昧性を持たせたことで、より人間に近い動きとなり、今まであきらめていた断定できないような判断に伴う業務処理についても、自動化できる可能性が広がります。推論エンジンには、いろいろな手法が使用されていますが、たとえば汎用性のあるアシスト社の「Progress Corticon」をRPA ツールと組み合わせれば、簡単にAIを搭載したような業務自動化が実現できます。

機械学習の発展

ルールベースの考えから発展してきたAIがある一方で、決定論的な柔軟性に乏しいという問題に対応するために、確率や統計の考え方を人工知能に導入して例外への柔軟な対応をとれるよう1985年に考案された「ベイジアン・ネットワーク」という手法があります。イギリスの数学者の「ベイズ理論」を応用したもので、まず一旦あたりをつけて主観的に確立を決めて、その現実との乖離を元に補正を繰り返すやり方です。

この方法では、徐々に初期確率を補正して精緻化するという学習方式をとるため、大量な学習データとそれを処理する強力なコンピュータパワーが必要でした。これに対し、1990年以降、個人向けPCも普及し、インターネットが爆発的に広がったことから大量のデータが容易に収集可能となりました。また、ベイジアン・ネットワークは、機械に学習させるという手法です。機械学習という手法は、ルールベースにも適用されていますが、Webとの親和性も高かったため確率・統計手法での存在感が際立っています。

機械学習の技法例

- ベイジアンネットワーク
- 決定木学習
- 帰納論理プログラミング
- クラスタリング
- ニューラルネットワーク
- 遺伝的プログラミング
- 強化学習

機械学習は、特定のアルゴリズムを指すわけではなく、実行することによって段階的に賢くなっていく手法であるといえるためいろいろな技法があります。事前にある程度方針を決め、集団別に分類することをゴールにしている「クラスタリング」や、学習成果を受けて報酬を与える「強化学習」などがあります。機械学習では、アルゴリズムの観点で、「教師あり学習」と「教師なし学習」、あるいは、「半教師あり学習」などにも分類されます。

そして、機械学習といえば、現在でも一番話題になっているニューラルネットワークです。神経が伝わる仕組みを活用した手法で、脳科学との関わり合いもあり、人の脳を模倣しようという考えは意外に古く戦前よりありました。その後、1986年に命名されたバックプロパゲーション（誤差逆伝播法）によってこれに「教師あり学習」を行なわせることで、機械学習により排他的論理和の問題などを克服しました。しかし、多層化することで学習の処理が複雑で、学習のさせ方自体が難しくなるのと、その計算負荷が膨大になりすぎるといった新たな課題も生じてきました。

ディープラーニングとビッグデータ

ニューラルネットワークによる機械学習の中でも、学習のさせ方が複雑で難しくなってしまうという問題に対して、答えを与えない学習を可能にしたということで、日々多くのメディアに取り上げられているのが「ディープラーニング」(深層学習)です。

ディープラーニングは、車の自動運転や音声検索等、人間の能力に依存していた画像認識や音声認識などの分野でめざましい活躍を見せています。

ディープラーニングは、単なる機械学習とは異なり、今まで人間が与えていた教師データなどの特徴の指示や、パラメータによるアルゴリズムの調整などをコンピュータ自身で行えます。人間の脳神経が3層以上あるように、ディープラーニングではニューラルネットワークを多層化して学習し、特徴となる要素をコンピュータ自身が抽出できます。よって、答えを与えない学習が可能のため、自律型のAIとして注目されています。



ニューラルネットワークの多層化で自己学習を可能にするためには、学習時間や処理にかかるコストが膨大になることが問題でしたが、GPUをはじめコンピュータ処理能力の向上によって近年実用化が可能になってきました。

技術の進歩はめざましく、たとえば現在のスマートフォンは、20年ほど前に、チェスの世界チャンピオンを負かしたIBM社のスパコンであるディープ・ブルーの処理能力に匹敵します。実際、すでにGoogleやFacebook等は、スマートフォン上でディープラーニングモデルを作成できるようにしています。

一方、ディープラーニングの大ヒットでビッグデータの役割、重要性がますます高まってきました。テクノロジーの発達によって、人間では到底計算できないような膨大なデータを元に答えを導くことができるようになり、今後この分野はさらに進化し続けるでしょう。



ディープラーニングに限らず、AIを搭載する場合、大量のデータを必要とするケースがあります。その操作を行うのに、高速でソートするための「Syncsort DMExpress」のようなツールや、高速解析のためにcsvの形式でSQL文を操作できる「解析ブースター」のようなツール等を利用すると、RPAが更に効率的に実装できます。

AI サービスと RPA

ビッグデータを活用するためにクラウドを利用したサービスが大変増えてきております。

AIも例外ではありません。

自社のノウハウを利用してAIを搭載する仕組みを構築するのであれば、そのノウハウが流出しないように自社のインフラでAIを構築するという選択もありますが、発展途上のAIをトリアル的に実装していくのであれば、クラウドサービスを利用する方法もあります。

いろいろな分野に特化したサービスが提供されているので、RPAで連携すれば割と手軽にAIを利用した業務が行なえます。

たとえば、AmazonのAWSなどのクラウドサービスでは、ディープラーニングを活用するためのフレームワークや、ライブラリーがあり、GPUなどの利用も可能です。新規の業務には有望な選択肢となるでしょう。

RPAで自動処理させる限り、ユーザーにはクラウドサービスを利用しているのかどうかなどは、あまり意識させずに使用することが可能です。

ここで注意すべきことは、音声認識、画像認識、テキスト解析などのAIテクノロジーをWeb APIの形で利用できるある大手のクラウドサービスは、企業向けオンラインサービスのプライバシーポリシーの中で、「Cognitive Servicesには適用されない」という文言が追加されたことが話題になりました。

ディープラーニングなど、AIサービスの提供業者は、自身のAI精度の向上を目的に、ユーザーがサービスの利用をする時のインプットデータをサービス業者側でも利用できるようにしているケースがあるということです。これは、サービス提供側の目的は理解できますが、ユーザーが個人情報等を自身で管理し、セキュリティ等についても自分達で責任をもってAIを利用しなければいけないとあらためて認識する必要があります。



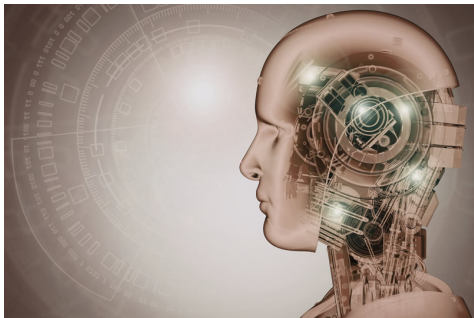
AI 搭載 RPA の今後

現在盛り上がっている RPA の実態は、まだまだ汎用的な AI は搭載されておらず意思決定が簡単な定例業務を中心にパターン化できる作業の自動化がほとんどです。一方、RPA のツールベンダーは、AI をどのように取り入れていくか模索中で、いろんな業種でそれぞれ AI 的に業務代行できるソフトウェアロボットを展開しています。

今後、AI が実際の業務に普及するためには、RPA を導入することがよい機会なのかもしれません。AI の取り扱いをソフトウェアロボットに任せることにより、人々はより高度な処理がなされたかのように勘違いし、処理も素早く正確なのできっと満足度も高まることでしょう。

AI を搭載することによる懸念点は、AI が出してきた答えが正しいものかどうかの検証が人間ではできなくなる場合が多いことです。つまり、AI が出してきた答えは、それがどのような推論であっても信じるほかなくなってしまうのです。ビッグデータを元に導き出した答えは、たとえそのアルゴリズムに問題があっても、人間では膨大な処理を検証する術もなく見逃してしまうことが多くなるでしょう。但し、それは致し方ないです。コンピュータの恩恵を受けるためには、今後 AI を搭載したシステムが不可欠になってきます。バックオフィスの自動化を進め、次に業務の改善、見直しをする場合には、必ずソフトウェアロボットにオペレーションを任せることを前提として自動化の改善をすべきです。

今後、どの業務にも役立つ AI が選別され使用できるように開発されれば、RPA ツールとして AI 標準搭載済みソフトウェアロボットが世に広まってくるでしょう。それまでは、業種や業態に特化した AI システムと有効な RPA ツールのソフトウェアロボットとをうまく組み合わせて、RPA の導入を進めていくべきです。



RPA による働き方改革

ソフトウェアロボットに頭脳といえる AI を組み合わせて RPA がそれらを実装できれば、働き方改革の推進に明るい未来が待っています。

事務担当者は、自宅からソフトウェアロボットの監視を行ない、時間があれば新たなソフトウェアロボットを自宅で量産し、遠隔地から業務をこなす時代が来るでしょう。

但し、典型的な RPA ツールはサーバ/クライアント方式の中央集中管理型を採用しているためロボットの運用のためにシステム管理者を別にアサインし、ロボットを管理する必要があり、冗長化やバックアップの問題を解決するためには、多くの投資が必要になってしまいます。

ROBOWARE のように、ソフトウェアロボット同士がリレー（中継）できると、PC で開発してそのままサーバ無しで IP ネットワークを通じて、他のソフトウェアロボットに指示が与えられます。人が仕事する場合と同様に、何か不都合があってもソフトウェアロボット同士で、代替してバックアップ体制をとることができます。コールドスタンバイとして、業務担当者のみで電源が入っていない別の PC を起動して、障害時も処理を継続させることが可能です。また、リモートデスクトップの環境などセキュアな仕組みが必要なテレワークでも、サーバ側で ROBOWARE を導入すれば、複数台のソフトウェアロボットをそれぞれ独立して動かすことができます。

2020 年からは、小学校でもプログラミングの授業が必須となります。

近い将来、プログラミングによって、ソフトウェアロボットを作成したり、AI のアルゴリズムをチューニングしたりすることは、日常の英会話を覚えるのが一般的になったのと同様、誰でも当たり前習得して普通に行われているだろうと予想されます。

将来の会社は、何台のソフトウェアロボットを自分の部下に持っているかで、給与や待遇が違って来る時代になるかもしれません。

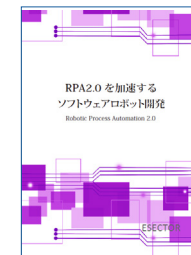
もうすぐソフトウェアロボットという働き者の部下が、私たちの仕事を楽にしてくれて、避暑地で余暇を楽しみながら働ける時代がきっとやってくるでしょう。

ソフトウェアロボットを使ってテレワーク！

RPA2.0 を加速する ソフトウェアロボット開発

2018年02月28日公開

<https://www.esector.co.jp/special/rpa/rpa5.html>



RPA2.0 とは？

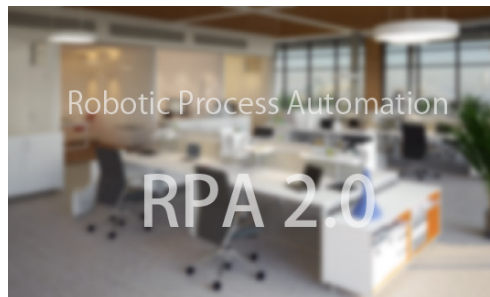
RPA がブームになり、バックオフィス業務の自動化が各企業で進行しています。その中でも、定型の作業だけでなく、人の判断を必要とするような複雑な業務についても自動化を実現できる RPA2.0 が、次世代の RPA として注目を集めています。

決まりきった PC の操作はロボットに任せてしまおうと考えるのは必然であり、それが RPA の推進を後押ししました。しかし、定型業務の自動化だけでは物足りない企業を中心に、業務担当者の経験や知識に依存していた業務に対しても臨機応変に対応できる RPA2.0 が、多くの企業から支持されて来ています。

広範囲の業務に対しても自動化するためには、テンプレートに当てはめるだけではなく、AI などを利用して様々なアプリケーションと連携して、自社の業務に最適なロボット開発をすることが、RPA2.0 として求められています。

RPA2.0 の特徴は以下の通りです。

- ①非定型な業務に対応できる
- ②適応出来る業務の範囲が広い
- ③現行の環境でそのまま使用できる
- ④ロボット同士の連携ができる
- ⑤メールデータなどの非構造化データを取り扱える
- ⑥複雑な条件分岐や、判断が行なえる
- ⑦いろんなアプリケーションと連携できる
- ⑧サーバでの集中監視が要らない
- ⑨最適な業務ごとのバックアップ体制が組める
- ⑩ AI や IoT と連携できる
- ⑪拡張性がある・・・など



企業内でのロボット開発競争

RPA2.0 が世間に認知され導入が始まると、サービス向上や生産性アップで同業他社に打ち勝つために、自社に有利になるように業務に適したより高度なソフトウェアロボットを開発しようという機運が盛り上がり、ますます開発競争が激化していきます。

ソフトウェアロボットは、開発系の RPA ツールを利用することでロボット開発の高度な知識がなくても、あらゆる企業や組織で割りと手軽に開発が出来るようになりました。

本来バックオフィス業務の効率化を目的にヒットした RPA ですが、デジタル労働者であるソフトウェアロボットに仕事を任せることで、あらゆる業務の作業効率と正確性が増すという期待で、フロント業務まで適用範囲を広げて開発が進んでいるようです。

企業の人材確保という観点で少子高齢化の問題を解決する最も有効な手段は、デジタル労働者であるソフトウェアロボットを有効活用することです。

65 歳定年制をはじめとする継続雇用制度の導入で高齢者も労働力として期待されていますが、体力や健康の問題で長時間勤務が難しい人達でも、ロボットに仕事を代行させれば、担当者は、ロボットへの指示やアプトブットの確認など、比較的無理のない仕事で活躍できます。

また、会社内でしか作業できない仕事をロボットに任せることにより、育児に追われる社員でも在宅勤務が可能になるため、優秀な社員をつなぎとめることにも役立つでしょう。

そして、何より事業を拡大したいのに、人材不足でスタッフを補充できないという問題も、ロボットの増員で解決が可能です。

ほとんどの企業は、事業継続、あるいは事業拡大のために、多かれ少なかれ今後労働者としてソフトウェアロボットを活用していくことは間違いありません。

むしろ、そうなる決まりきった仕事をロボットに任せるだけでなく、どれだけ多くの業務について、如何に効率的に作業できるロボットを開発していくかが、企業にとって生き残りの重要なポイントとなっていきます。



ツールとしてのロボット

企業がソフトウェアロボットを活用していく時に重要なのは、ロボットはあくまでもツールとして導入するという姿勢です。

AIを搭載したロボットは人間の仕事を奪うという恐怖を与えるような風潮がありますが、ソフトウェアロボットは、現時点ではあくまでも業務を自動化するためのツールです。

よって、業務担当者が拒否反応を示さないように、ソフトウェアロボットは自分達の作業を楽にしてくれるツールだということを理解してもらい、業務を担当する現場の人達が自ら主体となって進めていくやり方が理想的です。

家庭でも、全自動洗濯機やお掃除ロボットは、主婦の家事を助けてくれる良いツールで、それにより時間的余裕が生まれ生活が豊かになってきました。今では、洗濯機のない時代が考えられなくらい一般家庭に普及しています。

それと同様に、業務担当者にとってもソフトウェアロボットは有益だという意識付けが必要です。



注意すべきは、業務改革という大義名分によって経営者がAIを搭載したロボットは万能だと誤解して、コンサルティング会社の提案のまま担当者を無視して、トップダウンでソフトウェアロボットを導入してしまう場合です。この場合、ソフトウェアロボットの中の動きはブラックボックス化され、何か障害があっても担当者にはわかりません。ロボットの開発を依頼した業者に頼まないと、業務は動かなくなり、最悪はそういった外部の会社に自社の重要業務の生命線を握られてしまうことにもなりかねません。

危険な面もあるロボットですが、ツールであることを前提に活用すれば、業務担当者の主導で導入することにより企業に大きく貢献します。自動車を運転するのと同様、優秀であるがゆえにソフトウェアロボットも使い方や運用方法を間違えるとかなりリスクのあるツールであることを理解すべきです。

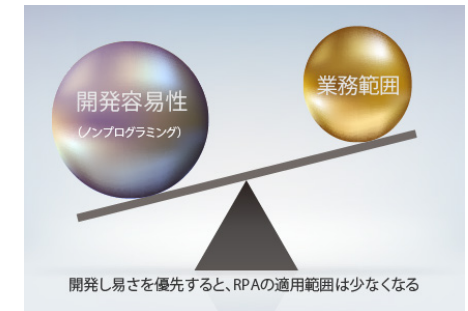
よって企業は、コストや、業務効率ばかり優先してトップダウンですべてをコンサルティング会社やRPAベンダー任せにするのではなく、危険でもあるが役立つツールとして業務を理解している現場の主導で、各担当も責任をもってソフトウェアロボットを導入することが、リスク管理をする上ではとても重要になります。

RPA ブームの問題点

RPAが大ブームとなったおかげで、多くの企業でいろいろなRPAツールの導入が始まりました。RPAへの期待が業務の自動化なのに、残念なことに多くのRPAツールはパターン化できる定型業務しか自動化できません。

RPAツールのGUI機能により、自動化する仕組みを作るのは比較的簡単になりましたが、そのかわり自動化できる業務は限定されてしまいます。

これはRPAに限ったことではありませんが、開発容易性の向上は、適用できる範囲を制限されることとのトレードオフなので、すべての業務を自動化するには、やはり無理があります。



結果的に、RPAで自動化したい業務範囲と、RPAツールが提供できる自動化が可能な範囲の差が、RPAを導入した企業の不満の原因になっているようです。

標準的なRPAツールは、分かりやすいGUIを使って、プログラミングなしで自動化の設定が簡単に実現でき、その設定した内容はフローチャートのようにドキュメント化できます。もちろん、こうしてビジュアルに設定できる方が、プログラミングが不得手な業務担当者としては抵抗なく業務自動化の設定ができます。

しかしながら、多くのRPAツールは、GUIによるマウス操作主体で設定できる自動化には限界があり、結果的に特定のスクリプト言語等を使用してカスタマイズしなければ、期待する業務の自動化ができないのが現状です。

ソフトウェアロボットは、パターン化できる決まりきった仕事しか自動化できないことを理解していればそれは仕方がないと諦めもつくのですが、定型のアルバイトでもできるレベルの業務であれば、コストや時間をかけてロボットを導入するメリットもかなり薄れてしまうのが実情のようです。

こうした問題が生まれた背景には、多くのRPAツールのベンダーが、他のツールと組み合わせで開発していたり、本来の機能とは別にかなり作り込んで構築したシステムを、さもRPAの成功事例のように記事広告にしているため、購入者がRPAツール単体でできることを拡大解釈して誤解したまま導入してしまうことも大きな要因となっているようです。

間違っていたロボット化のアプローチ

最近になって多くの企業が RPA の導入を始めたきっかけは、やはりメディア等で注目され、数多くの成功事例が世間で話題になってきたからでしょう。

実は PC の操作を自動化するツールは、数十年も前からありました。事実、RPA ツールの多くはかなり以前から販売されていたもので、近年になってブームに乗っかり RPA というキーワードにアピール方法を変更してから、売れ始めたものがほとんどです。

よって、多くの RPA ツールには特別先進的な機能の改良があったわけではなく、ブームが到来したおかげで売れていると言っても過言ではないでしょう。

ただ、企業の業務のロボット化の波は止められません。ここで後れをとれば、企業存亡の危機にもなりかねないからです。

多くの企業で、トップダウン的に RPA の導入が進められました。日本の大半の企業がそうであるように、とにかく失敗せず早く効果を得たいからと安心できる RPA ツール、つまり実績のあるツール、メディアで多く目にするツール、そしてコンサルティング会社が推薦するツールがどんどん採用されました。

採用したツールが、期待通りの業務範囲で自社の業務自動化が実現できればよかったです。その多くが ERP を導入したときと同じで、ソフトウェアの仕様に合わせてロボット化できるように業務フローを変更したり、一部分は自動化できず人による操作が残ったり、カスタマイズするために業者に依頼したり、あるいはプログラミング不要なはずなのに開発部分を加えたりしている企業がほとんどようです。

こうなることは、RPA のツールベンダーからすれば当たり前で、むしろそれを前提に提案しているため、勝手に期待したユーザの進め方にも問題があるといえるでしょう。

RPA ツールは、ソフトウェアロボットといえどもツールです。お掃除ロボットに洗濯を期待できないと同様、自社で期待する業務が、そのツールの標準機能でどこまで自動化できるのかそれを購入前に見極めてツールを選定すべきです。

もちろん、予算にもよりますが、業務ごとに最適なツールを複数使い分け、他のアプリケーションと連携して使用するという考え方も有効な手段です。



プログラミング技能の重要性

2020 年より、小学校でもプログラミングが必修科目になる予定です。

マイクロソフト、Google、Facebook はじめ名だたる有名企業の創業者の多くは、プログラミングやコンピュータサイエンスの分野に長けていました。

コードを書くプログラマーではなかったもののアップルの創業者だったスティーブ・ジョブズ氏も、オバマ前大統領と同じように、すべての国民がプログラミングを学ぶべきだと主張していました。プログラミングできる技能が、一般教養として求められる時代になりました。

なぜでしょうか？

AI、IoT が話題になり、ビッグデータの取り扱いやセキュリティが重要になってくるにしたがって、ロボット社会でコミュニケーションを図るには、業種を問わずプログラミングできる技能が必要だからです。

今後、ますます重要性を増し、拡大を続けるであろうソフトウェアロボットの導入にあたり、他人任せでは、会社で働く人達の将来が心配です。

プログラムコードの記述は開発の専門家に任せるともよいのですが、自動化するときの手順や、例外処理の対処方法、セキュリティや運用まで、プログラミングをする上での常識やノウハウを、経営者から一般の業務担当者まで関係者全員が取得していれば、その企業に合った理想的なロボットの開発が期待できます。

さらに、プログラミング不要ということで流行った RPA ですが、プログラミングすることを許容するだけで、適用出来る業務の範囲が広がり、人材不足の対応や、業務拡大のメリットを最大限に享受できる可能性が増えていきます。

海外とのコミュニケーションに、英語が重要であるように、今後ますます増え続けるソフトウェアロボットに指示を与え、報告を受け、共に業務を遂行する仲間としてコミュニケーションを図るためには、プログラミング技能を全社員が身に付けていくことが、どの業種の企業にとってもたいへん有効な手段となります。



AI への過大な期待

ロボットという、マンガに出てくるように賢くて何でもできる印象がありますが、ここでいうソフトウェアロボットと AI（人工知能）は同義語ではありません。

ソフトウェアロボットとは、PC をはじめとしたコンピュータをプログラムされた内容に従ってさも自律的に動かすソフトウェアのことで、ヒト型のハードウェアに搭載されるのではなく、PC 等に導入されて RPA を実現するデジタル労働者のことです。

AI は、人と同じような知能を有しているように見えるソフトウェアで、人型のロボット等のソフトウェアに組み入れられている場合もありますが、一般的にはロボットと連携していたとしても、まったく別物です。

ソフトウェアロボットが、手の操作にあたるキーボードやマウスなどを制御するのに対し、AI は脳で行われる判断などの処理を司るイメージです。

AI も適用範囲が限られているものが多く、話題のディープラーニングを活用している自律型の AI であったとしても、現時点ではまだ全ての行動や思考を自律して行える汎用人工知能（AGI: artificial general intelligence）のレベルにまでは、至っておりません。レイ・カーツワイル氏によって有名になったシンギュラリティ（技術的特異点）に人類が到達出来た時には、万能に近い AI が出現するかもしれません。



コグニティブが前提となっている将来の RPA3.0 では、RPA に予め搭載された AI が、ソフトウェアロボット自身に指示を出し、人の制御に頼らず自律して対応する RPA が期待されています。その頃には、ソフトウェアロボットの生産や修正も、ロボット自身が行なっているかもしれません。

いずれにせよ、現段階の RPA は、自身で思考できるわけではなく、あらかじめプログラミングしたソフトウェアが指示通りに動くだけです。

それゆえ、RPA2.0 では、RPA ツールが、個別に業務に適した AI と連携する方法でソフトウェアロボットを開発して、今まで人の判断が必要だった部分についても自動化します。

この部分をよく理解して最適な組み合わせを構築するためにも、一般教養としてのプログラミング知識や、AI リテラシーの向上が、経営層から一般社員に至るまで求められ重要となって来ます。

顧客満足度と社員満足度

ソフトウェアロボットの導入効果として、顧客満足度の向上が挙げられます。

ロボットへの指示に間違いがなければ、ソフトウェアロボットは人間が操作するより早くて確実に業務を遂行できます。

メディアで多く取り上げられた生命保険や宅配便の会社等で、チャットボットを利用したヘルプデスクの RPA の事例では、担当部門にロボットが導入され、実際のオペレータに交じってロボットが顧客に対応して、素早い対応と、オペレータの負荷軽減でかなり効果を上げています。これは、顧客満足度の向上と同時に、応対に余裕ができたことによる担当部門の人達の満足度の向上の両方に効果が上がる良い例です。

通常顧客満足度を上げるためには、現場の担当者がハードワークを強いられ、担当部門の不満増大にもなりかねません。

この例では、弱い AI で代表されるルールエンジンを用いたチャットボットを RPA ツールと連携して構築されています。以前より問題視されていたロボットが人間の仕事を奪うという心配は必要なく、このヘルプデスクの事例ではこれを導入することによって、現場の負荷軽減にも大きく貢献しているようです。

しかし、もし企業経営者の判断でこれをヘルプデスクごとアウトソーシングして、外部委託によってロボット化していたらどうでしょう。

人件費を中心としたコスト面の効率化はできるため、それなりの効果は見込めますが、ヘルプ業務でのノウハウは社内に蓄積できず、それが会社の発展にとってよいかどうかは判断が分かれるところです。

但し、このケースは、その業務を海外の人件費の安い BPO センターへアウトソーシングしてコスト削減する場合でも同じ事なので、必ずしもロボット化することが人の仕事を奪う原因とはいえません。けれども、RPA はもともと BPO センターの運営会社がコストメリットを出すために発展してきた歴史を知れば、ロボットが人の職場を脅かしているというのも事実ではありません。

RPA は、導入の仕方を間違わなければ、従来の顧客満足度を上げることができます。顧客満足度向上を優先すれば、コストと業務負荷を上げてしまうというトレードオフな関係ではなく、顧客満足度を上げながら、コストと業務負荷を下げ、社員の満足度も向上させることができるということが言えます。顧客も社員も経営者も三方満足する、これこそが、RPA のよい成功事例となります。



時短を実現するために

働き方改革を推進するためには、現在 8 時間以上拘束されている定時の勤務時間について、ロボットに業務代行をさせることで、業務に費やす時間を短縮することが望めます。そのためには、制度の変更や在宅勤務ができるようにセキュリティ面で環境を整える必要などもあるでしょうが、そもそもロボットを導入すれば物理的に業務担当者は勤務時間の短縮が実現できるのでしょうか？

残念ながら、RPA を導入したおかげで、勤務時間が半分になりましたという事例はまだ聞こえて来ません。これは、会社全体で時短を目的とすることが、明確になっていないためでしょう。どんなに業務効率がアップしても、企業の業務拡大を優先してしまえば、忙しいから増員したときと同じように、かえって仕事が増えてしまうかもしれません。

ここで重要なのは、ロボットを足らない人員の補充として採用するだけでなく、担当者の業務の一部の引継ぎ先として位置づけるということです。その際に、業務は引き継ぎますが、アルバイトを雇うときのように、責任は業務担当者に残すくらいがよいでしょう。そうすれば、ロボットをツールとして利用した担当者は、自分の仕事量が減って楽になるということで、そのまま業務で処理して出てきたアウトプットに責任を持ってくれます。そのように意識付けするには、業務担当者にロボット開発の時からメンバーとして参画してもらうことが重要です。

RPA ツールによっては、PC 自動化をサーバからの指示で管理されているため、業務担当者の独断では動かせない場合もあります。管理方法は会社の方針によりますが、理想はシステム管理者に依存せず、担当者が利用しやすいソフトウェアロボットがよいでしょう。たとえば、通勤ラッシュを避けるため、朝一にデータを集め処理しなければいけない業務は、ロボットに任せて出勤後に結果を確認できれば、時差出勤が可能になります。緊急で夜中に見積作成が必要な場合や、インフルエンザで会社に出勤できないような場合など、ファイルサーバにアクセスするためにどうしても会社での仕事することが必要なケースでは、担当者からリモートでロボットに仕事を依頼できればどんなに楽でしょう。

そのために、導入するロボットは、PC が立ち上がってなくても起動して、リモートで担当者自身が監視や制御ができることが理想です。このあたりの機能があるかどうか、RPA ツールを選ぶときのポイントになります。

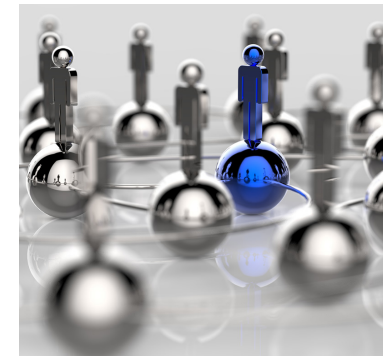


単独ロボットから連携ロボットへ

RPA (Robotic Process Automation) に似た言葉として、RDA (Robotic Desktop Automation) というキーワードがあります。これは、個別の PC にインストールして、マウスやキーボード操作を簡単に自動化できるもので、その多くは低価格ですが、スケジュールの機能がなかったり、リモートからの起動や監視ができなかったりします。業務担当者が手軽に作成できるのはよいのですが、簡単だからとむやみに担当レベルでジョブを作成してしまうと、会社としては管理が煩雑になり困るという悩みもあります。

ロボットに人の代わりをしてもらうのであれば、業務担当ごとにロボットを配置して役割分担が必要になります。そして、担当業務を分担して連携したり、作業終了後にアウトプットをデータ送信して、別の業務担当に渡して処理してもらうなど、人間で行なっているのと同じように連携して処理をすることができるのが重要です。

標準的な RPA ツールは、これをサーバで行なうものが多いのですが、集中管理する際の難点は、システム管理者がすべての業務を把握しているわけではないので、結局、不測の事態に対処する場合、今までよりも多くの人に介入してもらうか、業務権限の見直しをする必要性が発生してしまいます。各部門のみ権限が与えられている業務は、その部門でしかソフトウェアロボットが動かないようにする配慮が必要です。そうでなければ、部門ごとのアクセス権限等にも影響します。



また、サーバで集中管理する方式であると、PC のバックグラウンドでしか処理できないものがあったりします。逆に、RDA のようなツールでは、バックグラウンド処理はできず、フロント処理のみなのでロボット稼働中はマウスとキーボードが使用できないものもあります。業務内容によっては、画面を表示しながら実行した方が良い場合と、バックグラウンドで並行で高速処理したいものがあるので、両方が選択できることが望めます。さらにできれば、RDA のように PC 単独でも担当者側で動かすことが出来ながら、サーバに依存せず PC レベルで監視ができ、他のロボットと連携してスケジュール実行ができる仕組みが理想です。作成し易い優秀な RPA ツールでも、ロボットが人間のように柔軟に対応できないツールは残念ながら、RPA2.0 には向きません。

冗長化とバックアップ

ソフトウェアロボットは、人間のように風邪をひいたり病気になることはありません。但し、一切トラブルがないかという点、ソフトウェアロボットが入っている PC が故障したり、ウィルスに感染することで、ロボット自身が動けなくなる場合も想定されます。

こうした時、人間組織であれば基本は誰かが代わってその仕事をするバックアップ体制が整えられています。いつも A さんがしていた仕事を、A さんの具合が悪いので B さんが代わりに作業するような事は、一般的な企業であれば日常的に行なわれております。

ロボットに仕事を代行させる時も、この体制は必要です。A の PC に導入したロボットが起動できなければ、B の PC でも代行できるようにしておく、そして重要度によっては、人間による指示がなくても自動的に切り替わる仕組みが欲しいものです。



RPA ツールのほとんどは、この部分をサーバの冗長化の仕組み等に依存しており、システム管理者がいなければリカバリーできないなどの弊害もあります。ましてや、RDA などの PC 単独で動くツールでは、ほとんど冗長化する機能を持っていません。

その点、RPA2.0 に対応した開発ツールを使用すれば、想定されるリスクに応じた対策をプログラミングが可能のため、ホットスタンバイやコールドスタンバイなど、その業務に応じた冗長化をあらかじめ設定しておくことが可能になります。

ホットスタンバイまでは必要がない業務であっても、いざという時に備えて、2 体のロボットがお互いにバックアップとなるような仕組みなどは最低限用意しておくべきです。

多くの RPA ツールが、自動でリカバリーする機能を搭載していないため、トラブル時の対応についてユーザ側で仕組みを構築しなければなりません。この部分はソフトウェアロボットを運用するときに重要なポイントとなります。

その際、選択した RPA ツールによっては、機能不足により他のアプリと同じポリシーレベルのリスク回避基準での運用が無理な場合もあるため、あらかじめ購入前に確認が必要です。

多様性への対応

同業他社との差別化が、今後のロボット化時代に生き残るためには重要です。企業は、多様化する業務すべてに対して迅速に、そして柔軟に対応していく必要があります。そのためにも、パッケージ化された ERP や CRM のように他社でも使用されているアプリケーションに業務を合わせるという考えではなく、他社に比べ圧倒的にハイレベルな効率化が可能になるその企業独自のソフトウェアロボットを開発することが望まれます。

RPA は、自動化が目的のため、現在使用しているアプリケーションを代替するものではありません。ロボット化により、従来の業務アプリケーション等を、担当者が行なっているいつもの順番の通りにそのまま自動実行をさせることができます。これにより、担当者は、自動化のために業務フローを変更する必要がなくなるのです。

RPA2.0 では、あらゆるアプリケーション、あらゆるフォーマットのデータ、あらゆるセキュリティソフト等と連携することができます。

どのような業務にも対応できる理由は、たとえば開発型の RPA ツールである ROBOWARE の場合、C 言語のように機械語に近いプログラム言語を用いてソフトウェアロボットのフレームワークを作成しているため、CPU やメモリ、デバイスなど OS のカーネルに近い情報でも受け渡しを可能にしているからです。

但し、そのソフトウェアロボットへの指示は、フレームワーク上の API を利用して、Ruby などのスクリプト言語でコーディングできるので、高度なプログラミング知識がなくても、ロボットの開発ができます。

以前であれば、ロボットを作成するには機械語に近いプログラミング言語を用いて開発することが必要だったため、専門知識がないとかなりハードルが高かったのですが、今ではフレームワークにより、誰でもなじみやすい言語でロボットの開発ができるようになり、それが RPA2.0 の推進を大いに後押ししています。

プログラムなしで作成できる典型的な RPA ツールでは、パターン化できる決まりきった仕事しかできなかったのですが、Ruby や PHP などの開発が楽なスクリプト言語でプログラミングするところまで許容することにより、様々な変化や多様性に対応できる RPA を実現できます。



IoT とのネットワーク連携

他社の導入事例を参考に RPA を実施することは、リスク回避のためにもとても有効ですが、他社と同じレベルの自動化では、競争に負けてしまう可能性があります。

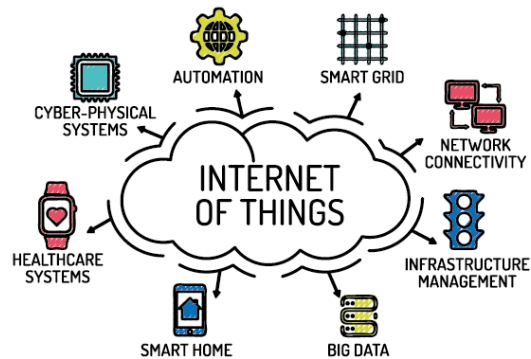
第 4 次産業革命の要は、AI と IoT だと言われておりますが、RPA2.0 は、その両方とを連携することにより、バックオフィス以外の業務でも適用範囲が各段に広がります。

PC 操作の自動化しかできないイメージであった RPA において、IoT から得られるセンサーの情報をコントロールしてアラートを出したり、IoT からの情報を AI につないで処理させたりすることで、今までできなかったような業務処理も実現可能になります。IoT の種類によっては、独自の管理コンソールを持っていますが、RPA2.0 は、他の IoT やアプリケーションまで連携して統合管理をすることができます。

バックオフィス業務で注目されている RPA ですが、システムの運用管理の目的で、データセンターやマシン室でも、大いに活躍します。

たとえば話題の AI 搭載のスピーカーと連動すれば、口と耳の代わりとなり、ロボットへの指示もマウスやキーボードではなく、声に代わるかもしれません。ロボット自身の監視も、異常があったら声や音で知らせてくれるでしょう。

RPA の当初の導入目的は、現状行なっている業務の自動化で充分でしょうが、RPA2.0 のノウハウが溜まれば、いままで人間の能力では実現できなかったような業務まで自動化をすることが可能になります。そこまでの領域に到達してこそ、ソフトウェアロボットは多くの人が期待していた賢いロボットとして活躍する時代となるでしょう。



RPA の未来

RPA2.0 の次は、自律型 AI によるコグニティブな RPA が期待されています。その頃には、AI を搭載した RPA が、自身でプログラミングしてロボットを量産し、ロボットをロボット自身が管理するロボット社会になるかもしれません。

そうなれば、業務は完全に人の手を離れ、担当者は業務の監視からも解放されます。

その頃には、ロボットをコントロールする能力がない人は、ロボットに管理されているかもしれません。その前に、ロボットに仕事を奪われているかもしれません。だからこそ、プログラミングの技能や AI リテラシーの向上が重要になります。

どんな世界になるかは、まだまだ想像でしかありませんが、今後各企業で RPA が推進されることにより、ソフトウェアロボットの開発が盛んになり、稼働するロボットの数が増えていくことは間違いありません。

その際、ロボットはパターン化した定型業務しかできないというメディアの洗脳にまどわされることなく、自社の業務に合ったロボットを開発して有効利用できた企業が、数年後でも生き残っていけるでしょう。そうした企業は、そのロボット開発のノウハウと実績でロボットを他社へ提供することで、その業界のイニシアチブを取ってビジネスを拡大させているかもしれません。

この流れは、もう誰にも止められないところへ来ています。

そうであるならば、手をこまねいているのではなく、むしろ 1 歩先を行く覚悟で、ソフトウェアロボットの開発にチャレンジしてはいかがでしょうか？



RPA の盲点 IT ガバナンスの重要性



2018年04月26日公開

<https://www.esector.co.jp/special/rpa/rpa6.html>

ガバナンスの必要性

RPA（Robotic Process Automation）の導入を考える時、どうしても優先して注目してしまうのが、デジタル労働者であるソフトウェアロボットの作り易さです。ロボットの作成には高度な知識が必要となり、とてもハードルが高いという懸念があるため、作り易そうな RPA ツールを第一優先にして選定する傾向があります。

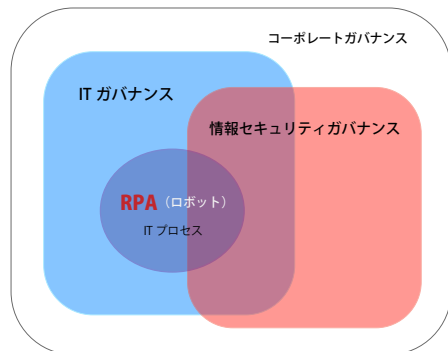
プログラムレスということが特長で、UI（ユーザインターフェース）を用いて比較的簡単にロボット開発が行なえる RPA ツールは、多くの企業や組織で採用されています。しかしながら、ロボットが停止したとしてもあまりインパクトが少ない業務であれば特に問題ないのですが、作成時の容易さよりも、ソフトウェアロボットを作成した後、どのように管理、運用していくかは、RPA にとって重要なポイントです。

企業として、IT ガバナンスは必要です。ソフトウェアロボットも例外ではありません。（ここでは「IT ガバナンス」は、財務会計に限定せず業務全般における内部統制の IT に関わる部分を指し、情報セキュリティのガバナンスも含んで説明していきます。）

RPA ツールによって、ロボットを簡単に作れるようになったからといって、今話題の「野良ロボット」のように管理できていないソフトウェアロボットが増えれば、セキュリティリスクは高まり、異常停止により業務に支障が出れば事業継続にも影響しかねません。

ソフトウェアロボットを、コンプライアンスやセキュリティ対策を考慮せず量産してしまっただけでは、かえって統制のための管理コストが増えてしまいます。コストが増れば、自動化したメリットは無くなってしまいます。

今後 AI と結びつきながら発展するソフトウェアロボットは、その最重要課題としてガバナンスが求められています。



ロボット運用の不安

RPA は、雑用レベルの単純作業から次のステージの RPA2.0 へ進み、定型業務以外の複雑な業務に対しても広がりつつあります。

RPA が個々の単純な PC 操作の自動化から、重要な業務に対して適用されるようになると、その効果と共に重要性が増し、業務における RPA の依存度の割合も高くなります。それ故、その処理が停止した時のインパクトも大きくなります。

- ・ロボットにパスワードを教えて大丈夫？
- ・画面のレイアウトが変わったらどうなるの？
- ・Windows のバージョンアップって関係する？
- ・どこの処理で止まったのか分かるの？
- ・暴走したりしないよね？

RPA では、デジタル労働者のソフトウェアロボットが、業務担当者に代わって PC 操作等の作業を実行してくれるため、そのソフトウェアロボットが止まれば業務も止まります。

RPA の処理の多くは、WEB ページや各種アプリケーションの表示画面の特定部分に対し、マウスクリックや、キーボードからのテキスト入力等を自動的に行ないます。

その処理の指示は、RPA ツールを利用して行うこととなりますが、処理対象であった画面のレイアウトが突然変わってしまった場合などは、予期せぬソフトウェアロボットの停止が発生し、即時に対応することが難しい場合があります。使用した RPA ツールですぐに新しい画面用にロボットを修正できれば良いのですが、多くの場合、業務続行を優先するため、停止した業務を手を使って対応することになってしまうのが実情です。ここで問題なのが、RPA の効果によって、人員削減まで行なってしまった業務は、いざという時に対応できる人がいないため企業に与えるダメージがかなり大きくなることです。

その対応に時間とコストを費やし人を雇うことになっては、本末転倒になりかねません。

企業にとっては、事業継続は死活問題なため、業務の異常停止は避けるべき大きな課題です。ソフトウェアロボットでの自動化を最適化するために業務フローを変更していた場合は、更に対応を難しくします。RPA ツールで処理させている部分が、ブラックボックスとなってしまうような動きをしているのかわからないため、以前のように人のオペレーションだけではうまくリカバリーできない場合もあります。

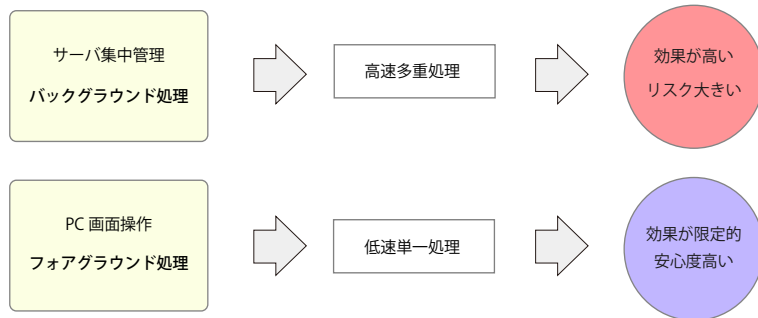
ソフトウェアロボット稼働の種類

RPA ツールによって作られるソフトウェアロボットにも様々な種類がありますが、統制の観点で分ければ、サーバでソフトウェアロボットを集中管理し仮想 PC 上でバックグラウンドで処理するタイプと、それぞれの PC 上で人の操作と同じように画面に表示してマウスやキーボード操作を行うタイプに分類されます。

バックグラウンドで処理できるサーバタイプは、マルチで並行処理できるため、高速な業務処理が期待できることと、集中管理が可能なのが利点ですが、PC 操作の処理内容が視覚的に見えないため、動きがブラックボックス化されてしまい、エラー時の対応が専門知識を持つものに依存するというリスクがあります。開発をベンダー任せにしている場合など、即時対応が難しくリカバリーにコストもかかります。

一方、個々の PC でフォアグラウンド処理するソフトウェアロボットは、画面上で自動操作されている動きが見えるので、異常時のエラー箇所が特定し易く、いざという時は、人が代わってオペレーションすれば対応できることが多いという安心感があります。但し、こちらの方式では、現場の業務担当が簡単にソフトウェアロボットが作成できるタイプが多いので、リスクとしては、「野良ロボット」を量産し易いという懸念があります。

但し、開発型の ROBOWARE のように、バックグラウンドの処理でも、画面を見ながらのフォアグラウンドの処理でも、適正に合わせて両方の使い分けが可能なツールもあります。しかも、サーバがなくても PC でジョブのリモート管理ができ、業務に合わせてカバナンスの目的でログ出力を追加するなど、複雑なコーディングもできます。



誰がロボットを管理するのか？

ソフトウェアロボットを作った人が、そのロボットを管理すれば良いのでは？と思われがちですが、IT ガバナンスの観点からすれば、それがベストであるとは言えません。

RPA ツールでソフトウェアロボットの作成を現場の業務担当が行なった場合、IT 部門と比較すれば決してシステム技術やセキュリティ対策のノウハウなどの知識が高いわけではないので、作ったソフトウェアロボットがセキュリティホールになる可能性さえあります。そのため、パスワードなどは、ロボットに入力させない方がよいという意見もあるくらいですが、いちいち担当者がパスワードを入力しているのでは、中途半端な自動化になってしまい、意味がありません。

では、開発を委託した IT ベンダーにセキュリティ面も含めてトータルな管理を任せられるかという、コスト的にもかなり高額になり、業務の機密に関わる部分の漏洩リスクと、エラー時の対応が遅くなるという心配があります。更に社内にノウハウが残らないため、外部の会社に業務を握られてしまうという経営リスクにもなります。

ソフトウェアロボット
誰が管理すべき??



- 現場の業務担当?
- システム部門?
- RPA ベンダー?
- RPA 専門チーム?
- CLOUD アウトソース?

それでは、RPA の専門チームを作成して管理運営を行なえばどうでしょうか？

ソフトウェアロボット開発の段階から、RPA ツールの使用方法を学び、作成したロボットを会社全体で統括して運用できるようにするわけです。この体制は、理想的ですが、システム部門は現場の業務に精通していないため、業務部門と、システム部門の両方からメンバーを選出し構成することになります。大きな会社組織であれば可能かもしれませんが、自動化のために人をアサインするにはかなりコスト面の負担を覚悟しなければならなくなります。

但し、今後ロボットは加速して増えていくので、将来全社的にソフトウェアロボットに業務を任せられる場合には、この RPA 専任チームが、ロボットに業務を代替してもらった人達が所属すべき職種となるかもしれません。このように、ロボットが仕事を奪うというよりは、ロボットを利用して、仕事が高度化していくのです。

ロボット管理の進め方

ソフトウェアロボットにおいて、理想的な管理はどのような方法で行なうのでしょうか？ RPA 専任チームもよいのですが、そのために人的コストをかけたのでは、自動化によってかなりコスト削減ができる仕組みが必要であり、スモールスタートから始める早期段階では、メリットどころか RPA によって経費が増大してしまいます。

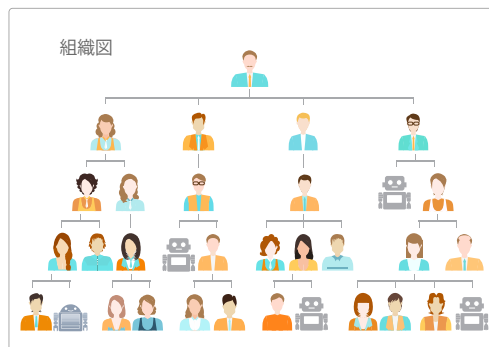
そこで現実的なのが、ソフトウェアロボットを人として管理する方法です。ソフトウェアロボットは、そのそも人の作業の代行をします。多くの企業や組織では、すでに所属する人に対して、ガバナンスの体制が出来上がっています。デジタル労働者であるソフトウェアロボットも同様に、人と同じように ID、パスワードを持ち管理するのです。

たとえば、営業部で働くソフトウェアロボットは、その業務を行う人の PC と同じ Windows ID でログオンさせれば、営業部で認められた権限範囲で作業を行なえます。つまり、経理部など他部署の業務を動かしたり、そのファイルサーバのデータを読み書きすることはできません。

いずれにせよ、自動化したい業務を稼働していた PC でソフトウェアロボットを動かす、そのロボットには、その PC で行っていた人の持っていた権限以上の権限を与えなければ、その業務担当者に任せることができます。任された担当者も、もし、ソフトウェアロボットが他部署のファイルを暴走して壊すことができる権限を持っていたら、絶対そのロボットに責任を持たたくないでしょう。

複数ソフトウェアロボットを管理する場合は、その業務を行なっている責任者に委ねることができます。たとえば、ROBOWARE であれば、サーバでなくても上長の PC に Job Manager を

入れることができるため、いつものデスクから指示や確認ができます。A さんが突然休んだから、B さんに仕事を頼むように、不測の事態では、他の人の PC で稼働させることも、ソフトウェアロボットに同じ部署内の権限を与えておけば可能になります。



運用リスクのマネジメント

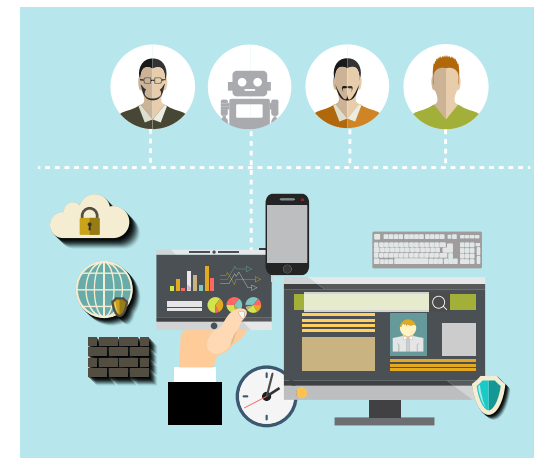
業務の処理内容に関しては、その業務を行う部門でハンドリングできますが、ソフトウェアロボットはソフトウェアであるため、システム部門の関与も必要になります。

IT ガバナンスはシステム部門が重要な役割を担い、すでに確立されている企業が多いでしょう。ソフトウェアロボットはそこに組み込む管理対象のソフトウェアとして、当然ながらセキュリティや品質管理が求められますが、それに加えてファイルサーバに対するアクセス権限など、システムを守るために、人と同じ扱いで管理対象に組み入れるべきです。

業務部門だけでは解決しない障害が起これば、システム部門の介入を要請するわけですが、これは他の重要アプリケーションと同様に、システムのヘルプ担当に社内標準ツールとして、サポート対象に組み込んでもらう必要があります。RPA ツールが、様々なアプリケーションに連携するため、システム部門で管理している OS などのバージョンアップや、パッチ情報の管理と同様に、使用した RPA ツールのバージョン管理も加え、影響があれば予防保守など事前に対処する仕組みが必要です。

システム部門がどこまでソフトウェアロボットのの使用に関する権限を与えるかについては、ソフトウェアロボットを擬人化して考えれば、現状の IT ガバナンスの中で、各業務部門に対してシステムが持っている権限を参考に設定すればよいのです。

そう考えれば、ソフトウェアロボット用に新たに障害時のフローを作成するというよりは、現状の障害時の対策フローを基準にソフトウェアロボット特有の部分を検討するようにアレンジして追加修正をする方が、IT ガバナンスのためにも比較的スムーズに適用できます。



RPA とセキュリティ

RPA は、スモールスタートで段階的に、計画的に導入する方がよいとされています。産業用ロボットを工場に一斉導入する場合と同じように、数百台のソフトウェアロボットを導入して集中管理で業務を一挙に自動化する方法もありますが、それでは、障害時の対応がかなり大きなリスクとなります。

まずは業務担当者の PC にソフトウェアロボットを導入し、段階を経て業務を引き継ぐ方法が現実的です。いきなりバックグラウンドでの処理を行なってしまうと、正常に終了したと思っても、途中の過程の処理が間違っていないかどうかのチェックが難しくなります。そのため、ロボットによる自動化の操作も担当者と同じ PC の画面に動きを表示させて、それぞれのステップが正常稼働しているかを確認し、実績が出てからバックグラウンドへ移行します。

業務担当者には、それぞれの担当者が所属する部署のアクセス権限があり、たとえば営業事務の人が、経理の業務データにアクセスできてはいけません。ソフトウェアロボットについても、Active Directory 等でファイルのアクセス管理をすべきなのは当然ですが、そもそも営業部から、経理部のソフトウェアロボットで実行指示ができてしまうと、セキュリティ上の脅威となります。RPA ツールによっては、ソフトウェアロボット同士の通信機能やリモート制御の機能がないものもありますが、リモート通信ができて、セキュリティ基準が違う他部署への実行指示が出来るようなソフトウェアロボットの運用は絶対行なってはいけません。

この点について、例えば ROBOWARE の場合、ソフトウェアロボットをリモートで動かすためには、相手先の PC の ROBOWARE のロボット ID を知っている必要があります。このロボット ID は、厳格に管理され、第三者にコピーされないようにするためにパスコードによってレジストリに登録し、配布したロボット ID のファイルは削除できます。レジストリに登録されたロボット ID の情報は、コピーされても他の Windows OS では動きません。



通常のアプリケーション管理に加え、このようにソフトウェアロボットのリモート操作に対するセキュリティ対策は、RPA を導入する上でとても重要になります。ROBOWARE のような開発型のツールであれば、ソフトウェアロボット開発時のコーディングに様々なセキュリティ対策の処理を組み込むことができますが、サーバ集中管理を行う RPA ツールの場合は、管理者が特権を持ってしまうため、その RPA ツールが企業の業務分掌におけるセキュリティポリシーの基準を満たせるかどうか事前に確認することが重要です。

パスワード設定のリスク

RPA では、いろいろなアプリケーションを連携することが多いため、そのアプリを自動起動するためには、アプリ毎に ID、パスワードが必要となります。そもそも、Windows OS でさえ、それを設定しなければ立ち上がりません。しかしながら、RPA ツールの自動入力の設定に ID、パスワードを登録しておくセキュリティ上とても危険だということが指摘されています。

たとえば、記録型 RPA ツールは、手順としては画面上での操作をそのまま登録するため、ID やパスワードはテキストのまま記録されてしまいます。これでは、あまりに危険なため、少なくともパスワードは、セキュアなアクセス制御された場所にある暗号化されたファイルから、実行時にその都度入手する方法などへ変更しなければなりません。

これに対応するためには、企業の環境によりパスワード管理が汎用的ではないので別途スクリプト等でプログラミングしなければならない場合が多く、記録型 RPA ツールの中にはパスワード入力の部分は自動化を推奨していない場合もあつたりします。セキュアなパスワード設定の自動化ができないような RPA ツールを選択するのは、人手の介入が必要となり中途半端な自動化でしかないため注意が必要となります。

その点、ROBOWARE のように開発型 RPA ツールであれば、企業環境にあったコーディングが可能であるため、様々なアプリケーションを連携するような複雑な業務でも、ID やパスワードの情報をセキュアな環境から入手するような作り込みが行なえます。

これに付随して、たとえばリモートからの実行指示については、新たなパスワードのチェックを加えるなど、業務の重要度や、企業のセキュリティポリシーに合わせて、ソフトウェアロボット開発の段階でリスク対応計画に即したセキュリティ対策を組み込むことができます。



エラー処理の重要性

リスクマネジメントに必要なのは、想定されるすべてのリスクを洗い出し、それに対しリスク対応計画を作成し、リスク発生に備えます。同様に、ソフトウェアロボット開発においても、自動化する業務で想定できるすべてのエラーケースを洗い出し、それが起こった時の対処を事前に設定しておくべきです。ソフトウェアであるソフトウェアロボットは、業務エラー、システムエラー両方に対し、原因がわからず異常終了しないように、可能な限り業務継続できるように、通常のアプリケーション開発で行なっているのと同様以上の配慮が必要です。

RPA では、たとえばマウスクリックの対象にしていた画面について、アプリ側の都合で突然アイコンやバナーが変更された時などは、たとえ人間であれば認知して対応できるような場合でも、ソフトウェアロボットは停止してしまいます。また、想定外のデータが入力された時や、ファイルが見つからない場合もエラーになることが多く、対応が必要です。

RPA ツールによっては、GUI でエラー処理も設定するため、開発のし易さとのトレードオフになるため、条件分岐の数や設定できる内容を制限しているものが多いです。

画面レイアウトが変わった
おかしなデータが入っていた
前処理が終わっていない
通信できない
Windows アップデートが入った…



雑用レベルの簡単な業務であればそれほど問題視する必要はないかもしれませんが、複雑な対処が必要な処理を行なう場合、GUI で設定できる RPA ツールでは、エラー分岐が複数存在すると結果的に RPA に何を処理させているのかわからなくなり、かえって作業効率が低下する懸念もあります。

その点、ROBOWARE のような開発型の RPA ツールは、アプリケーションに依存することなく多種多様な環境でも動作し、エラー分岐による生産性の低下も少なく複雑な処理も設定できます。サンプルスクリプトでも日本語でコメントが書かれており、開発の段階でエラー処理を追加する場合も、どのような処理の時に必要な処理かを同じように分かり易くコメントしておくことで、異常発生時のソースコードの修正も迅速に行えます。

リスクマネジメントでは、エラーが発生したらその事象と対処した内容を知識ベースに追加して、できれば、次回発生した時には業務が止まらないよう改善できることが理想です。そのためにも、ソフトウェアロボットの開発において、ポイントごとにアプリケーションログを残すようなコーディングを手間を惜しまずしておくことが、不測の事態が起こった時の迅速な対処につながります。また、起こってしまったエラーは、ソフトウェアロボットに対処方法を組み込み、2度と停止しないように修正しておく必要があります。

ディザスタリカバリ

RPA を導入しても、当然ながら事業継続マネジメントを考慮したソフトウェアロボットの運用が必要です。

ソフトウェアロボット自身が、他のロボットで代替がきくように、バックアップ/リカバリーができる仕組みはもちろんですが、障害時のフェイルオーバー等の仕組みも必要です。

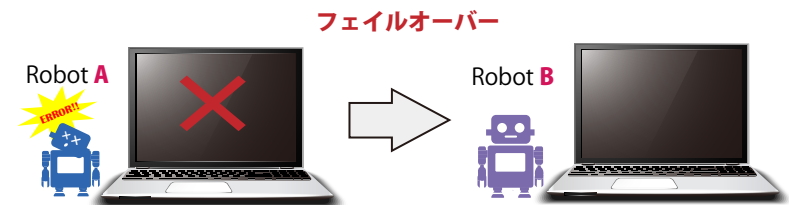
多くの RPA ツールは、障害時、災害時など不測の事態に対して標準で対策できる仕組みが弱いので、導入前に確認することが重要です。

サーバ集中管理型は、その中央管理のサーバに問題が発生すると業務に対するインパクトが大きいため、通常は管理サーバ用のハードウェアを2台以上複数使用し、クラスタリングして冗長化するなどの対策が必要になります。

PC 側でも障害は発生するかもしれません。ロボット A の PC がダウンしてしまった場合、ロボット B で作業をさせたいのに、ロボット B の PC に電源が入っていなかったため切り替えられなかったということでは意味がありません。

そういった対策として、ROBOWARE であれば、ソフトウェアロボット同士のバックアップができるため、サーバがなくても、PC レベルでもリカバリが可能です。また、バックアップ機の PC の電源が入っていないなくても、ネットワーク越しで通信するので、Wake-on-LAN 機能を使用できる環境であれば、マジックパケットを送信して、リモートで Windows を立ち上げ、ソフトウェアロボットを動かすことができます。

さらに、人が介入するウォームスタンバイに加え、フェイルオーバー機能として災害時に自動切換えを行なうホットスタンバイも設定できます。



RPA の導入時には、運用費用として企業で策定された基準にあったディザスタリカバリに関わる費用も考慮することが大切です。冗長化やバックアップの機能が不足している RPA ツールでは、このために二重投資が必要になったりして、かえって費用が増えてしまうこともあり得るため注意が必要です。

バージョンアップへの対応

ソフトウェアロボットもソフトウェアですので、バージョンアップを伴います。RPA の目的である業務自動化においては、様々なアプリケーションが関連するので、そのアプリのバージョンアップにも対応が必要な場合があります。

Web ブラウザのバージョンアップや、Windows OS のバージョンアップなどについては、ソフトウェアロボットを止めないためにも、事前検証できるテスト用の PC があると安心ですが、最近ではセキュリティ対策として自動更新してしまうアプリケーションも多いため、そういったケースに即時対応できる環境が必要です。

GUI でスピーディに作成できることも大切ですが、作成したソフトウェアロボットを修正し易いことも重要です。また、内容によっては、RPA ツール自身のバージョンアップを待たないと対応できない場合も想定されます。その点、海外製品であると日本語環境でのエンハンスのプライオリティが低くなってしまふ恐れがあり、国産のツールの方が安心だと言われています。

ソフトウェアロボットの開発に使用している言語にも注意を払わなければいけません。多くの RPA ツールは、ノンコーディングで開発できることを特長としておりますが、複雑な分岐や特殊な処理は、スクリプト言語を使用しているものがほとんどです。ROBOWARE のように、Ruby や PHP など複数の汎用スクリプト言語を選択できるのであれば、影響は少ないのですが、通常はその RPA ツールで特定しているスクリプトを使用します。

たとえば、古くから実績のある RPA ツールで多く使用されている Microsoft 社の VBScript の場合、脆弱性が懸念されており、Microsoft Edge ブラウザには対応しない方針で、Internet Explorer 11 でさえ VBScript の利用は推奨されておらず、標準のドキュメントモードである“IE=11”モードでは動作しない状況です。

こうしたスクリプト言語に頼っている場合、自社の PC 環境を維持するシステム部門にも、脆弱性対策のためのバージョン追従方針等に影響がでます。



誰がロボットを作成すべきか？

RPA ツールは、比較的簡単に定型業務の自動化ができることで普及したため、業務担当者による開発の試みが多く企業で行なわれました。

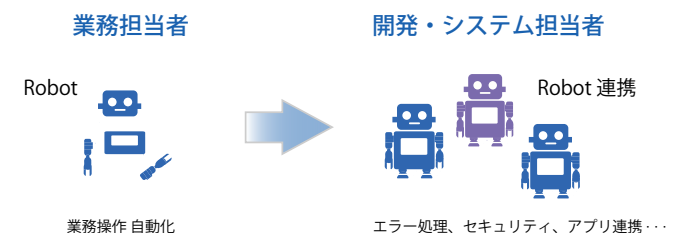
しかしながら、プログラミング知識が無くては複雑なエラー処理に対処できず困る上に、安易に現場でロボットを量産すると、「野良ロボット」が生まれてしまう問題もありました。

IT の統制を考慮した場合、業務担当の個人レベルでのソフトウェアロボット作成は控えた方が良いでしょう。各部門で、組織に基づいた権限をソフトウェアロボットにも与え、社員同様コンプライアンスなども配慮してロボットの作成が必要です。ソフトウェアロボットはソフトウェアですから、セキュリティポリシーに従い管理できるようにソフトウェアロボットを作成します。たとえば、各タスクの重要なポイントでは、業務処理のログを別途出力し、システムのエラーも収集して、ログ管理システムに繋げる仕組みも必要でしょう。

そうなれば、システム部門からの支援も必要です。企業規模によっては、ソフトウェアロボットの開発をする人と運用の担当は分けて進める場合もありますが、多くの RPA ツールはサーバ集中管理型なので、システムに精通した人が RPA 専任でソフトウェアロボットの開発をすることが期待されます。

結果、システムに精通して IT ガバナンスをよく理解している人が、業務担当者と連携して作成する形が現実的であると言えます。業務担当が一番業務のオペレーションを分かっているのですが、業務フローやマニュアルがしっかり作成されている場合は、それを元に業務担当とは別の人がソフトウェアロボットを開発することも可能です。

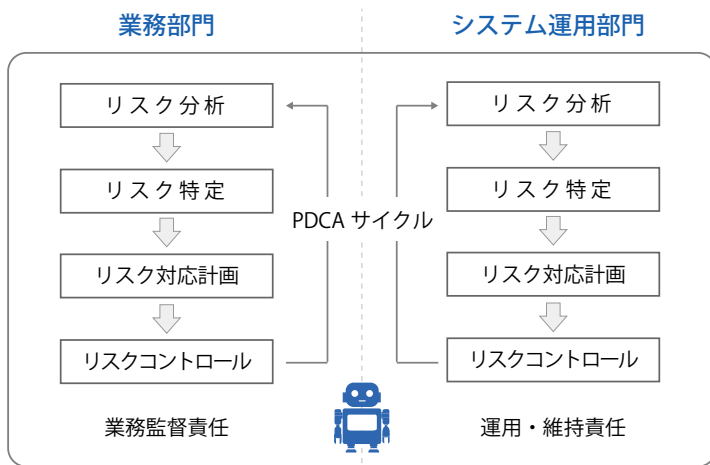
たとえば、ROBOWARE であれば、オプションの QuickROBO が操作記録型で自動化できるので、業務操作部分のみ現場で作成して、エラー処理やセキュリティ対策、アプリ連携などはシステム側で開発するという連携が可能です。



ソフトウェアロボットの責任所在

それでは、誰がソフトウェアロボットに責任を持つべきでしょうか？
 業務をロボットに代行してもらうのですから、業務によってはアウトプットされたデータ等を確認する作業があるはずで、通常は、その確認を行っていた今までの担当者が、ソフトウェアロボットを監督し、その業務担当部門が業務遂行について責任を持つべきです。

一方、ソフトウェアであるソフトウェアロボットには、バージョンアップや、障害対応などシステムの一部として管理すべき内容があります。業務寄りの部分は、業務部門に管理を任せ、ソフトウェアの維持という点では、システムの運用部門が、適宜開発部門の協力を得ながら全社的に管理した方が効率が望ましいです。



それぞれの責任範囲を明確化し、リスクのマネジメントが必要です。リスクを洗い出し、対応計画に従い、リスク事象が発生していないかを監視し、変化があった場合は、PDCA サイクルにて対応を改善していきます。

ソフトウェアロボットが導入されている PC に障害があると、当然業務に影響があるため、インフラの面でも注意が必要です。PC などの資産管理については、全社で一括管理している場合と、部門ごとで管理している場合があるので、それに準じて PC やネットワークなどの障害についても、責任を持って対処することになります。

IT ガバナンスには、当然コンプライアンスも重要ですので、RPA ツールのライセンスの管理や、セキュリティを意識したソフトウェアとしての資産管理が必要です。

ガバナンスと企業の将来

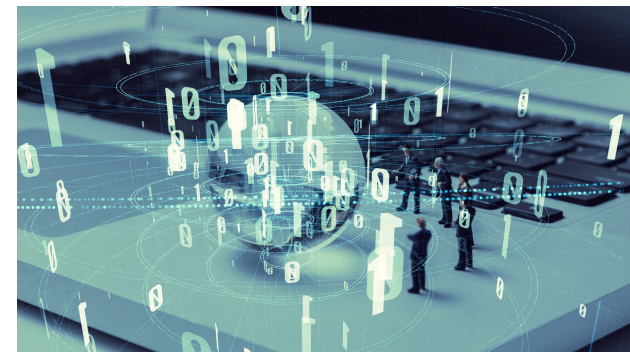
今後、企業にとってソフトウェアロボットの開発は日を追うごとに加速し、ますます過熱気味に拡大していくことでしょう。RPA による業務自動化対応に遅れをとっては、他社に先を越され、企業にとって存亡の危機となりかねません。

初期段階から IT ガバナンスを意識して RPA を導入した会社と、そうでない会社は、RPA 適用範囲が増えれば増えるほどその差は歴然となることでしょう。IT の統制がとれていない会社は、ソフトウェアロボット量産によって事業継続におけるリスクがとてつもなく増大することになります。

アニメ映画のようにロボットが暴走してしまうようなことは考え難いですが、悪意を持った第三者にリモートで制御され、予期しない動きや、あるいは知らない間に機密情報を流出してしまったりということも考えられます。

現実的に起こりそうなのは、順調に自動化が進みソフトウェアロボットにより人員削減され、業務担当の大部分が会社からいなくなった後、忘れたところにソフトウェアロボットが異常停止してしまう場合です。こうした状況は、大規模な業務改革とコスト削減を狙い、RPA ベンダーや、コンサルティングファーム等の提案をトップダウンでそのまま採用してしまった場合に想定でき、社内人員だけではリカバリできない場合は最悪な事態となるでしょう。

巷ではロボットや AI が仕事を奪うことが心配されていますが、正確にはロボットが仕事を奪うのではなく、ロボットをコントロールする人間が仕事を奪うのです。そのことを理解し、外部からの提案にすべてを依存するのではなく、ソフトウェアロボットに関するリテラシーを高め、そして社内にノウハウを溜め、ソフトウェアロボットを開発していくことが重要です。



この資料は、2017年1月31日より公開されたRPA(Robotic Process Automation)概説シリーズを転記し編集したものです。

<https://www.esector.co.jp/special/rpa/rpa-index.html>

2017年01月31日公開	RPA 概説①	RPA って知っていますか？
2017年06月13日公開	RPA 概説②	RPA 導入のために知っておきたい豆知識
2017年07月04日公開	RPA 概説③	ソフトウェアロボットの作り方
2017年09月27日公開	RPA 概説④	AI搭載を見据えたRPAの実装方法とは？
2018年02月28日公開	RPA 概説⑤	RPA2.0を加速するソフトウェアロボット開発
2018年04月26日公開	RPA 概説⑥	RPAの盲点 IT ガバナンスの重要性

※記載されている会社名、製品名またはサービス名は、各社の商標、または登録商標です。

CCS 株式会社 シーイーシーカスタマサービス

ESrabbit@cec-ltd.co.jp 〒150-0022 東京都渋谷区恵比寿南 1-5-5 JR 恵比寿ビル 8F
(企画・制作：春日井) TEL：03 (5789) 2443 FAX：03 (5789) 2575

