

ITO 現場における RPA ツールの活用事例

Case Study of RPA Tool at ITO Site

大 川 健

要 約 ユニアデックスでは、顧客のシステム基盤の構築から保守運用までを請け負う IT アウトソーシング (ITO) サービスを提供している。この ITO サービスの品質改善を目的として、Robotic Process Automation (RPA) ツールを ITO サービス現場の GUI 作業に試用した。「SI 保守サービス」における障害の検出と復旧処理の自動化を試みたところ、GUI 操作を含む想定した処理プロセスを自動化することができ、インフラエンジニアの作業負荷を軽減する手段として、RPA ツールが有効であることを確認した。

Abstract Uniadex provides IT Outsourcing (ITO) Service that undertakes not only building but also maintaining and managing customers' system infrastructures comprehensively. To improve the quality of the ITO service, we decided to use a Robotic Process Automation (RPA) tool for GUI operations of the ITO service site. We tried to automate detection and recovery processing for "System Infrastructure (SI) maintenance service". As a result, it was found that the automation of the target process including GUI operations was possible using the RPA tool, which made us confident that the RPA tool is quite useful to reduce the workload on IT infrastructure engineers.

1. はじめに

IT アウトソーシング (以下、ITO) サービスは、ICT インフラトータルサービス企業であるユニアデックス株式会社 (以下、ユニアデックス) の主要事業領域の一つである。ユニアデックスが提供する ITO サービスは、長年培ってきた経験を基に、複数ベンダーのソリューションやプロダクトが混在する IT 基盤環境の構築から運用までをトータルに引き受け、ワンストップで対応することを目指している。また、更なる高品質化と効率化を目指して、新しい技術要素やサービスの調査、適用にも積極的に取り組んでいる。

昨今注目が高まっている技術要素に Robotic Process Automation (以下、RPA) がある。RPA は業務自動化技術であり、近年多くのサービスプロバイダーがサービス提供コストの削減、サービスレベルの向上に向けて取り組みを積極化し、ユニアデックスも ITO サービスの様々な領域において RPA の導入検討を開始している。

本稿では、ITO サービス提供現場の課題に対して RPA を適用し、その効果を検証した事例を取り上げ報告する。2 章で RPA の概要、3 章で事例を説明し、4 章で ITO の現場における有効性をまとめる。

2. RPA とは

RPA とは「ロボットによる業務プロセスの自動化」である。但し、ロボットといっても、工場内で単純作業に従事する産業用ロボットではなく、人がコンピュータの画面上で行うク

リック、コピー、テキスト入力などの、いわゆる GUI 操作での作業を代行する「ソフトウェアロボット」のことを指している。単純なルーチン作業は工場だけでなく事務所にもあり、事務的な定型作業を自動化するのが RPA である。これまでロボット化が進んでいたのは、製造ラインの現場が中心だったが、RPA はその適用範囲をホワイトカラーの事務的な業務領域へと拡大するものとして期待されている^[1]。

2.1 RPA 活用のメリット

RPA を効果的に活用することにより、以下のようなメリットが享受できると考えられている^{[1][2]}。

- ・業務時間の短縮
- ・人為的ミスの撲滅
- ・コンプライアンスやセキュリティの向上
- ・24 時間 365 日の業務継続 (Digital Labor)

RPA がここ数年注目を集めている背景には、少子高齢化に伴う労働力減少、ホワイトカラーの間で蔓延する長時間労働に対する社会の目線の厳しさ、などの社会情勢の変化がある。労働力人口の減少をカバーするために、女性労働力の活用や雇用延長、未就業者の就業支援、外国人労働者の受け入れといった対策が講じられているが、人手不足が解消されないため、その打開策として、RPA が注目を集めている^[1]。

2.2 開発 API 型 RPA と録画テンプレート型 RPA

RPA には様々なソリューションが存在し、その分類の仕方もいくつかある。ここでは開発 API^[1] 型 RPA と録画テンプレート型 RPA という分類について紹介する。

開発 API 型 RPA では、C# や Ruby などの汎用的なプログラミング言語をベースに、GUI 操作の API を使用してソフトウェアロボットを作成する。これに対し、録画テンプレート型 RPA では、人間の GUI 操作そのものを録画する、あるいはマウスクリックやテキスト選択などの操作をあらかじめ用意されたテンプレートを使ってロボット化することで、ソフトウェアロボットを簡単に作成できる。

録画テンプレート型 RPA の方が直感的な操作でソフトウェアロボットを作成でき、プログラミング経験がなくても利用できるというメリットがある。しかしその反面、処理可能な操作がテンプレートに依存するため、開発 API 型 RPA に比べて複雑な処理の操作の自動化には不向き、というデメリットがある。開発 API 型 RPA と録画テンプレート型 RPA のどちらを選択するかは、ソフトウェアロボット開発者の IT スキルや、自動化対象業務の内容を総合的に勘案して決定することが望ましい。

3. ITO サービスの現場で抱えている課題に対する RPA 適用検証

ユニアデックスでは、ITO サービスの様々な領域において RPA の導入検討を開始している。本章では、いくつか試行した事例の中から「SI 保守サービス」への適用検証結果を紹介する。

3.1 SI 保守サービスの概要

SI 保守サービスとはシステム基盤を構成する各プロダクトの製品保守ではなく、システム基盤全体における重大障害対応と問い合わせ対応（設計内容含む）をサービス内容とするシステム基盤保守サービスである。現状では、業務アプリケーションを保守対象外としているため、システム保守ではなくシステム基盤（System Infrastructure：SI）保守と定義している。

3.2 自動化検証のターゲット

インフラエンジニアの作業負荷軽減を図るために、RPA による自動化検証のターゲットとして、以下の二つの作業を選定した。

- (1) VDI*² ログイン不可障害の検出と復旧処理
- (2) 構築および運用フェーズにおける負荷テスト

3.2.1 VDI ログイン不可障害の検出と復旧処理

一つ目のターゲットは、ユーザがVDIにログインできなくなる障害が発生した際の検出と復旧作業である。これまでは、ユーザから仮想PCにログインできない旨連絡があったタイミングで、運用オペレータがGUIベースでVDIを手動復旧するという対応をしており、作業工数を要するとともに、ユーザがVDIを使えない時間が発生することで、サービスレベルの低下を招いていた。

この事象に関しては、運用上デバッグモードでの原因調査ができないことから、原因の特定に至っていない。しかし、仮想PCがログインできなくなる条件をある程度掴めており、その条件に基づいてVDIを復旧するという一連の作業プロセスをRPAで自動化できれば、ユーザが気づく前に障害の未然防止ができるため、RPA活用の検証ターゲットとした。

3.2.2 構築および運用フェーズにおける負荷テスト

二つ目のターゲットは、負荷テストである。システム基盤構築時の負荷テストに利用できる、最大ユーザ数分の同時負荷がかけられる適切なツールとなるか検証した。

3.3 検証に利用した RPA ツールの概要

今回行った二つの検証には、オルタフォース社のRobowiserをコムスクエア社がOEM提供しているパトロールロボコンという製品を利用した。パトロールロボコンは、主にソフトウェアロボット（Robowiser Framework Server）とAPI、Windowsアプリケーション画面の属性情報の取得支援ツール（Application Analyzer）から構成されている。

パトロールロボコンは、この製品自体が人のオペレーションを自動化するものではなく、基本的には人がすべてプログラミングする開発API型という位置づけの製品である。但し、2018年7月現在の最新版（RBF_Professional-1.1.3.4）では、録画形式でプログラムを自動生成するQuickRoboという新機能が搭載されている。

本検証にあたっては直感性の高いUIが必須ではなく、むしろGUI操作部分以外の自動化は通常のプログラミングで柔軟に実装したいため、開発API型のパトロールロボコンを採用した。

3.4 GUI操作のプログラミング概要

GUIをプログラムで制御するには、デスクトップ上に表示されている対象アプリケーションの画面（ウィンドウ）の属性情報を用いる。Application Analyzer で画面の属性情報を取得し、ライブラリ（RBF_BaseAPI）に含まれている FindWindowHnd メソッドに渡して、制御したい画面のウィンドウハンドル ID を取得する。取得したウィンドウハンドル ID を操作対象の画面識別子として利用し、ライブラリとして提供されている GUI 操作メソッド群（マウスクリックやキーボード入力、クリップボードへのコピー等）を使って自動化したい操作を記述する。

開発言語として、Java, Ruby, PHP, C# がサポートされており、Application Analyzer で操作対象画面の属性情報を取得した後、「Copy」ボタンをクリックして、開発言語を選択すると、メソッド群に渡す引数（操作対象画面の属性情報を含む）が選択した言語用に自動生成され、そのまま開発環境にペーストして利用できるようになっている（図1）。

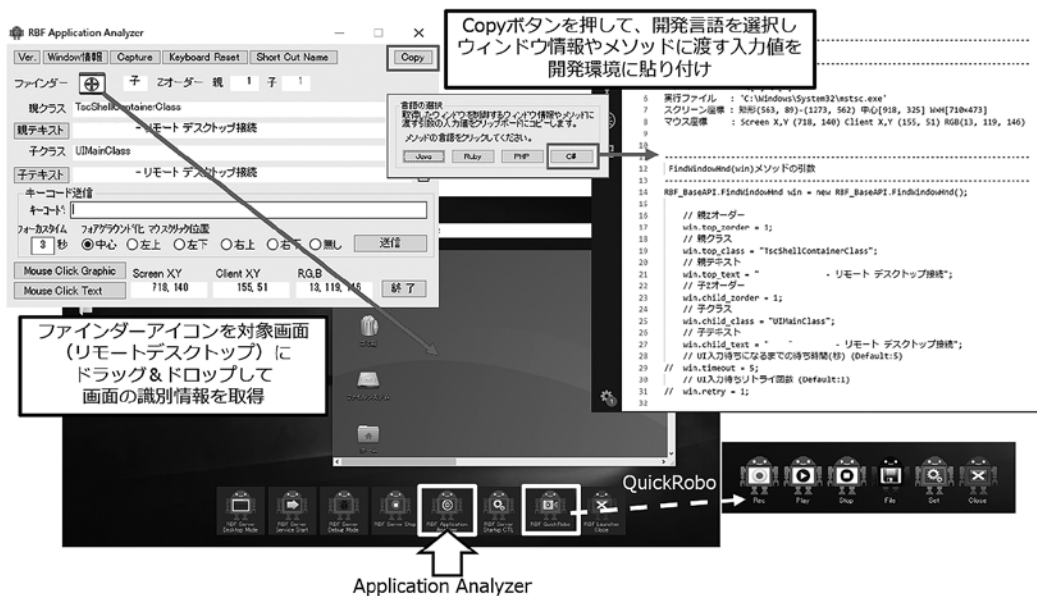


図1 パトロールロボコンの Application Analyzer を使った画面情報の取得イメージ

3.5 検証結果その①— VDI ログイン不可障害の検出と復旧処理プロセスの自動化—

一つ目の検証として、VDI ログイン不可障害の検出と復旧処理の自動化について報告する。

3.5.1 障害検出と復旧処理プロセスのシナリオ

VDI ログイン不可障害の検出および復旧処理プロセスのシナリオは以下の通りである。

1. バッチファイルを実行し、仮想マシンのステータスを管理するデータベース (SQLServer) に接続する。
2. SQL コマンドを発行し、VDI セッション最終切断時間から1時間を超過しても残っている仮想マシン (vCenter からのセッション終了コマンドに VMware Tools が反応してい

ないことから、何らかの理由でハングアップしており、この状態でユーザがログインを試みると失敗し、ログイン不可になると推測)の仮想マシン名を抽出する。

3. 抽出された全仮想マシン名をファイルに出力する。
4. ファイルに出力された仮想マシン名を取得する (検出件数が 0 件であればそこで終了)。
5. vSphere クライアントを起動し、vCenter サーバにログインする。
6. vCenter 管理画面上で該当仮想マシンを検索し、検出された仮想マシン名を右クリックしてパワーオフを選択する。

ステップ 3 の処理で検出された仮想マシン台数分繰り返す。

3.5.2 自動化検証環境

図 2 に示す仮想基盤上に検証環境を構築し、RPA ツールと vSphereClient、そして開発環境として VisualStudio を管理コンソールサーバにインストールして検証を行った。

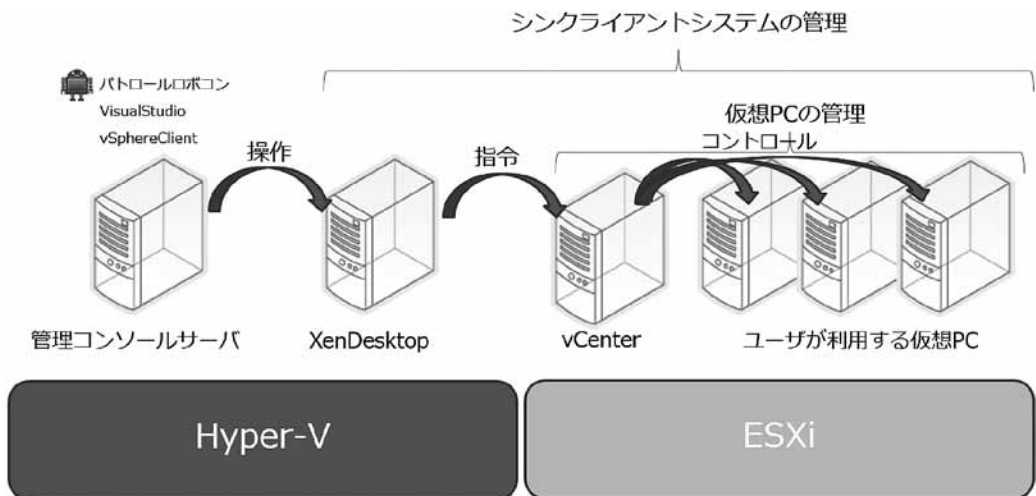


図 2 検証環境構成

3.5.3 障害検出と復旧処理プロセスの自動化

作成したプログラムの処理フローは以下の通りである。

■ログイン障害マシンの検出

- (1) エクスプローラを起動する。
- (2) エクスプローラのウィンドウサイズを調整する。
- (3) バッチファイルの保存フォルダに移動する。
- (4) バッチファイル (sql.bat) を実行する。
- (5) ログイン障害マシンが検出された場合、ファイルにホスト名を出力する。

■ここから復旧処理開始

- (6) vSphereClient を起動する。
- (7) 認証画面で IP アドレスを入力する。

- (8) 認証画面でユーザ名を入力する。
- (9) 認証画面でパスワードを入力する。
- (10) ログインボタンをクリックする。
- (11) vCenter へのログイン処理が終了するまで待機する。

■ 1 台目の障害マシンをパワーオフ

- (12) vCenter 管理画面上のインベントリ検索ボックスに、バッチ処理で検出された障害マシン名を入力し、検出されたコンピュータ名をクリックする。
- (13) インベントリパネル画面に該当マシンが表示されるので、そこをクリックする。
- (14) 該当マシンをパワーオフする。
- (15) ポップアップするパワーオフ確認ダイアログで「はい」ボタンをクリックする。

■ ステップ(12)～(15)を繰り返し、2 台目以降の障害マシンをパワーオフ

- (パワーオフした仮想マシンは vCenter の管理機能により自動的に再起動される)
- (16) vSphereClient を閉じる。

本検証の結果、作成したプログラムをソフトウェアロボットが実行することで、VDI ログイン不可状態になっていると想定される仮想マシンを検出し、その仮想マシンの復旧処理を行うために、vSphereClient を起動し、認証処理を行い、vCenter にログインして、該当マシンを再起動する、といった障害マシン検出から復旧までの一連の処理プロセスを、人の手を介さず自動化できることを確認した。

この自動化によって、ユーザからのログイン不可障害コールに対応する負荷の削減が期待でき、更には、ユーザが気づく前に、仮想マシンのログイン不可状態を検出して復旧処理を済ませ、ユーザへの影響を未然に防ぐことも期待できる。

3.6 検証結果その②—構築および運用フェーズにおける負荷テスト—

二つ目の検証として、RPA ツールを使った負荷テストについて報告する。

3.6.1 リモートデスクトップ接続に伴うクライアント端末側のリソース消費

想定ユーザ数で同時接続した状態のサーバ側負荷を計測するにあたり、接続元となるテスト端末側の必要リソース量について確認した^{*4}。リモートデスクトップの画面サイズが 1366 x 768 の場合、一つのリモートデスクトップ接続（接続先マシンの OS 起動状態、アプリ起動なし）で 150MB 程度のメモリが消費された。このことから、500 ユーザの同時接続をテストしたい場合は 75GB、1000 ユーザの同時接続をテストしたい場合は、150GB のメモリがテスト端末側に必要ということになる。実際にこれだけのリソースをクライアント端末側で確保することは容易ではないため、現実的な負荷テストのシナリオとしては、RPA ツールを使ってリモートデスクトップ接続し、シナリオに基づく操作を実行後、サーバ側のプログラムを動かしたまま切断する、というプロセスを想定ユーザ数分繰り返すという手順とした。

3.6.2 リモートデスクトップ同時接続プロセスの自動化

以下に、Windows Server 2016 にリモートデスクトップ接続し、切断するという処理を想定ユーザ数分繰り返すプログラムの処理ステップの概要を、パトロールロボコンの API を一部明記したかたちで示す*5。

- ① RoboticAPI クラスのインスタンスを生成し、roboapi という名前の変数に割り当てる。
- ② ソフトウェアロボット (Robowiser Framework Server) に接続する。
 - ②-1 roboapi の Open() メソッドに渡す引数のインスタンスを生成し、connect という名前の変数に割り当てる。
 - ②-2 引数 connect を設定する。
 - ②-3 ソフトウェアロボットに接続する。
- ③ ユーザ認証情報が格納された CSV ファイルを読み込む。
- ④ CSV ファイルの末尾まで読み込んだか確認して、その結果に基づき処理を分岐する。
 - ④-1 ファイルの末尾に達していない場合、ファイルからデータを 1 行読み込む。
 - (1) ファイルから 1 行分の文字列を読み込み、その文字列をカンマ区切りで分割し、logon_info という名前のリストに格納する。
 - (2) リストに格納した三つの要素を、それぞれ COMPUTER_NAME, LOGON_USER, LOGON_PASSWORD という名前の変数に割り当てる。
 - (1 人目のユーザ名を標準出力に表示する)
 - (3) リモートデスクトップを起動する。
 - (4) 「リモート デスクトップ接続」画面の認証処理を行う。
 - (5) 「Windows セキュリティ」画面の認証処理を行う。
 - (6) 「リモート デスクトップ接続」画面の認証処理の続きを行う。
 - (7) 次のユーザでリモートデスクトップ接続するために、現在のリモートデスクトップ接続を切断する。
 - ④-1 ファイルの末尾に達していないので、ファイルからデータを読み込む。

・・・ファイルの末尾に達するまで、ステップ④の繰り返し
- ⑤ 所定のユーザ数分のリモートデスクトップ同時接続処理が完了した旨、標準出力に表示する。
- ⑥ ソフトウェアロボット (Robowiser Framework Server) への接続を終了する。

この複数ユーザによるリモートデスクトップ同時接続プロセスの自動化によって、サーバ側からは実施できないユーザ認証を介したプロファイルに紐づく想定最大ユーザ数分の仮想マシンの起動とその負荷計測ができた。

4. ITO の現場におけるインフラエンジニアの負荷軽減手段としての RPA ツールの有効性

本稿で紹介した VDI ログイン不可障害の検出と復旧処理プロセスの自動化ならびに複数ユーザによるリモートデスクトップ同時接続プロセスの自動化検証を通じて、GUI 操作が必須の作業を自動化する手段として、RPA ツールが有効であることを確認できた。

一つ目の検証では、作成したプログラムをソフトウェアロボットに実行させることによって、障害マシン検出から復旧までの一連の処理プロセスを、人の手を介さず、自動化できるこ

とを確認した。この障害でコールを受けた場合、1) ユーザからの電話を受けてインシデント登録、2) 本番環境へのログイン、3) vCenter から対象 VDI を再起動、4) インシデントのクローズ処理、といった一連の対応に 20 分程度要し、同様のインシデントが月に 40 件前後発生しているため、年間で約 40 人日の対応工数が生じていることになる。障害検出と復旧処理の自動化プログラム作成に要した工数は 3 人日程度なので、RPA ツールを活用することで、障害コール対応工数の削減、更にはユーザへの影響の未然防止が期待できる。

二つ目の検証では、リモートデスクトップ同時接続プロセスの自動化によって、サーバ側からは実施できないユーザ認証を介したプロファイルに紐づく想定最大ユーザ数分の仮想マシンの起動とその負荷計測ができることを確認した。社内で共通利用できる負荷テストツールがない場合、負荷テストツールの作成に多大な労力を要する。対象となるシステム基盤の規模にもよるが、一例として最大 1000 人のユーザが利用する VDI システム基盤の場合、負荷テストツールの作成に 2 人月（エンジニア 2 名 × 1 ヶ月）程度要した。今回用意した負荷テストツールの作成に要した工数は 3 人日である。この負荷テストツールを共通基盤として展開することで、案件ごとに個別に対応してきたインフラエンジニアの負荷テストツール作成工数の大幅な削減が期待できる*6。

以上の結果から、インフラエンジニアの負荷を軽減する手段として、RPA ツールは有効であり、ITO サービスの品質改善ツールとしても期待できる。

5. お わ り に

世の中の RPA の動向としては、IPA（インテリジェントプロセスオートメーション）へと進化を遂げる次世代 RPA なるものが謳われており、画像認識や文字認識、自然言語処理などのコグニティブ技術や BPM（ビジネスプロセスマネジメント）技術が RPA に取り込まれ、2018 年から 2019 年にかけてトライアル期間を経て実用段階に入ると予想されている^[3]。

こういった RPA の最新技術を柔軟に取り込み、他の RPA ツールも活用しつつ、引き続き負荷テストの検証を進め、負荷テストツール、負荷テスト手法を確立する。そして長期的には、判断を含む作業の自動化を見据えて、SI 保守サービスを含む ITO サービスの効率化と高付加価値化を目指す。

最後に、本稿で取り上げた検証にご協力いただいたインフラエンジニアの皆様、ならびに SI 保守サービスをよりよいものにしていくための活動に共に取り組んでいるシステム基盤保守プロジェクトの皆様がこの場を借りて御礼申し上げる。

-
- * 1 API は、Application Programming Interface の略で、アプリケーションを開発する際にプログラマが利用できるように、あらかじめ提供されている開発済みプログラムのことであり、プログラムを利用する際の仕様を記載したドキュメントを含む。
 - * 2 VDI は、Virtual Desktop Infrastructure の略で、クライアント PC のデスクトップ環境をサーバ上の仮想マシンとして提供する仕組みであり、ユーザはクライアント PC からネットワークを通じてサーバ上の仮想マシンに接続し、デスクトップ画面を呼び出して操作する。
 - * 3 Windows アプリケーションを持つウィンドウ領域の中に含まれる描画可能な領域をクライアント領域と言い、このクライアント領域の中の位置を示す座標がクライアント座標である。クライアント領域の左上隅を原点 (0, 0) とする。これに対し、PC モニタの左上隅の点を原点とする座標をスクリーン座標と言う。パトロールロボコンでは、操作したいアプリケーションの画面サイズを調整する際に、スクリーン座標を利用し、マウスクリックする場所を指定する際にクライアント座標を利用する。

- * 4 社内にある Windows Server 2016 Datacenter の検証サーバにリモートデスクトップ接続した際に消費されるリソース量について確認した。使用したクライアント端末は、Dell LATITUDE E7240 (Microsoft Windows10 Pro, Intel Core i5 2GHz 2コア, メモリ 4GB) である。
- * 5 本番環境で想定される RDS サーバのセッションブローカ認証等には追って対応予定である。
- * 6 本稿で取り挙げた二つの検証とは別に、GUI 操作を必要とする端末キッティングにも RPA ツールを適用し、作業負荷軽減に有効であることを確認している。

- 参考文献**
- [1] RPA BANK, 【決定版】RPA (ロボティック・プロセス・オートメーション) とは?, 2018 年 5 月, <https://rpa-bank.com/report/7318/> (URL 確認: 2018 年 8 月 24 日)
 - [2] 田村直也, IBM THINK Business, RPA (ロボティックプロセスオートメーション) による定型業務からの解放, 2017 年 3 月, <https://www.ibm.com/think/jp-ja/business/rpa> (URL 確認: 2018 年 8 月 24 日)
 - [3] 野村総合研究所ビジネス IT 推進部/NRI セキュアテクノロジーズ, IT ロードマップ 2018 年度版, 東洋経済新報社, 2018 年 3 月, P36 ~ 37

執筆者紹介 大川 健 (Ken Ohkawa)

2007 年ユニアデックス(株)入社。前職の海洋調査会社では、観測データのデータベース化と海洋汚染物質の拡散予測に従事。2013 年から統合システムマネジメントサービスにおける予見化サービスの企画開発に携わり、その延長線上で 2016 年からシステム基盤保守サービスの企画開発に従事。現在、ITO 戦略推進部基盤技術課に所属。

