

ワークフロー開発実践テクニックレシピ
～第1弾～
ワークフロー動作安定化

2020/07/20

UiPath株式会社

ソリューション本部

セールスコンサルティング第3部



免責事項

- 本資料は情報提供及び弊社製品の利用をご検討いただくことを目的とするものです。目的外のご利用はご遠慮ください。
- 本資料に記載されている情報については、弊社では細心の注意を払っておりますが、その正確性や妥当性を保証するものではありません。
- 本資料に記載されている情報は、参考情報として提供されるものであり何らの法的拘束力を有するものではありません。
最終的な契約条件は、当事者間で有効に締結される契約において確定されます。
- 本資料に関する一切の権利は弊社に帰属します。本資料に関する著作権は、弊社又はその他の権利者に帰属しており、著作権法その他の法令で保護されています。上記の目的を超えて、著作権者及びその他の権利者の許諾を得ることなく、これらの情報を使用(複製、改変、転載などを含みます)することは、著作権法により禁止されております。また、本資料で使用される会社名、製品の商品名及びロゴマークは、弊社、そのグループ会社、又はそれぞれの権利者に帰属している商標又は登録商標です。権利者の許諾を得ることなくこれらを使用することは法令により禁止されておりますので、事前に当社にご連絡の上許諾を得ていただくようお願いいたします。

Contents

0. はじめに

0-1. 対象者（RPAデベロッパー基礎プログラム修了者）

0-2. 本資料で提供する価値

0-3. ワークフロー動作不安定時の解決フロー

1. セレクター認識が安定的にできているが動作が不安定な場合

1-0. セレクター認識が安定的にできている時の解決方法優先度の考え方

1-1. 画面描画前にロボットが動いてしまっている（ロボット処理が早すぎる、アプリが遅い、ネットワークが遅い）

1-2. 要素が有効になる前に動いてしまっている（少し遅れてボタンが押せる）

1-3. （無駄な）ある要素がランダムで表示される

1-4. ある処理の終了を適切に待てない

1-5. アプリ起動に失敗する

2. セレクター認識が安定的にできておらず動作が不安定な場合

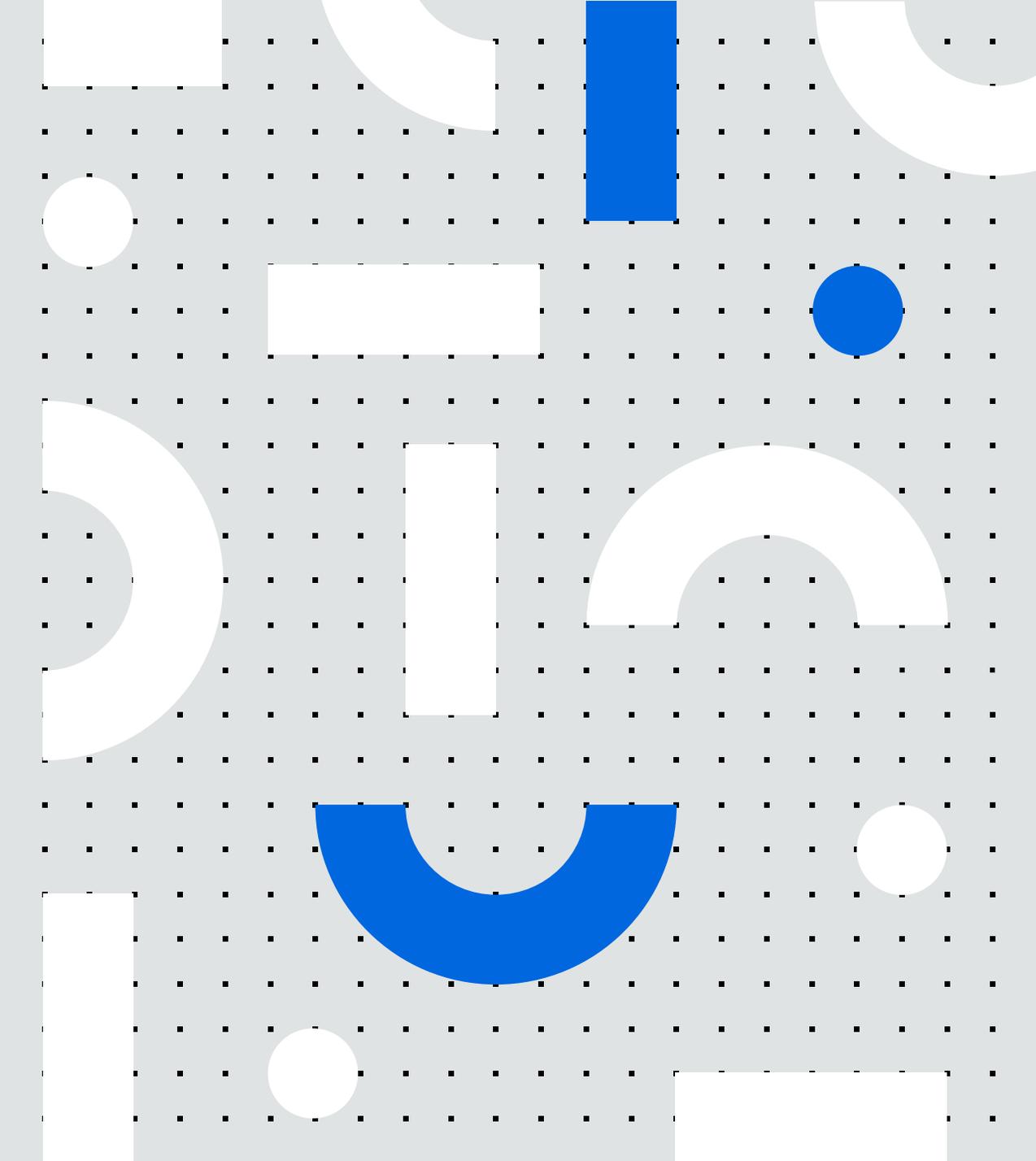
2-0. セレクター認識が安定的にできていない時の解決方法優先度の考え方

2-1. セレクター認識に不安がある（たまに変わる）

2-2. セレクター認識ができなかったため画像認識を使っているがたまにエラーが出てしまう

3. Appendix

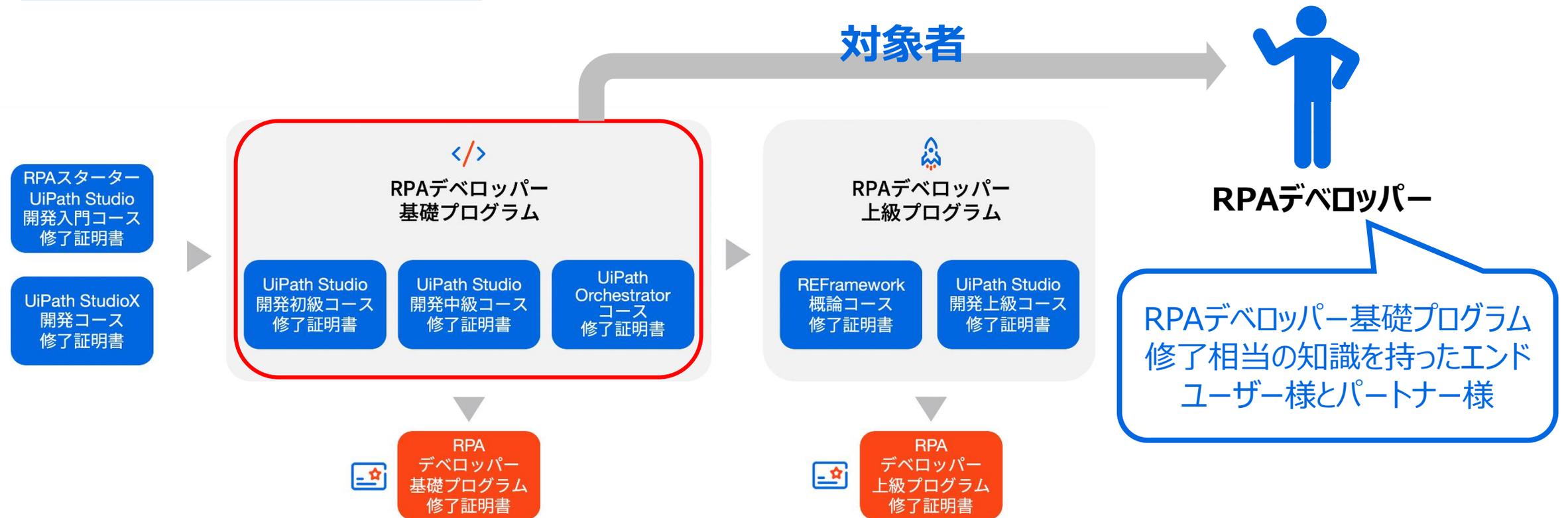
0. はじめに



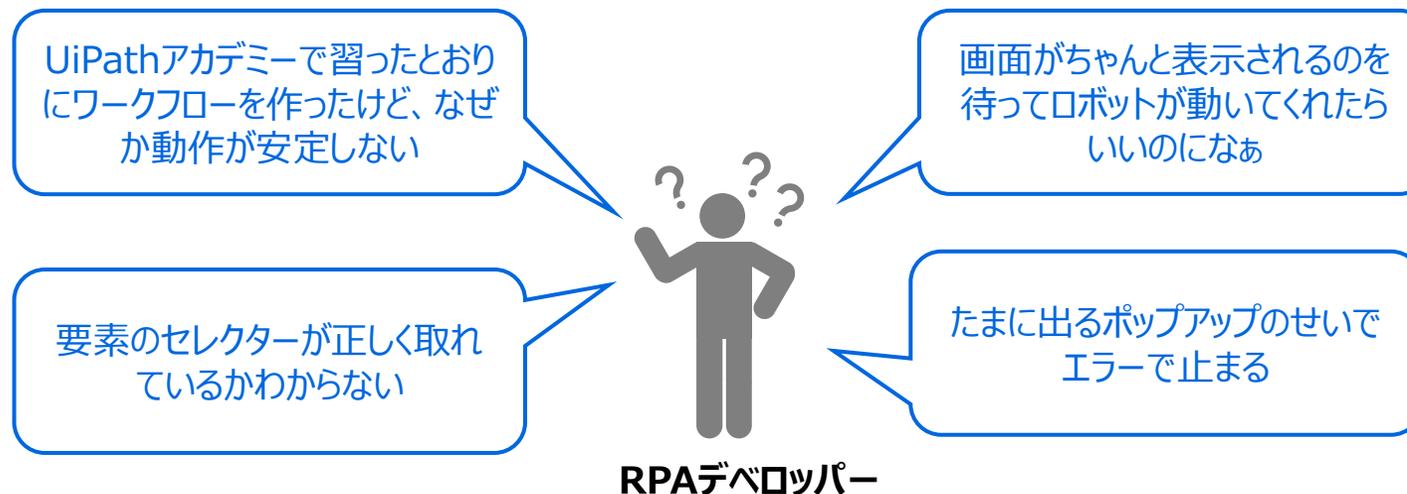
0-1. 対象者（RPAデベロッパー基礎プログラム修了者）

UiPathアカデミーにおける学習パスのRPAデベロッパー基礎プログラム修了相当の知識を持ったエンドユーザー様とパートナー様が本資料の対象者となります。

UiPathアカデミーにおける学習パス



ワークフロー開発時に、ワークフローの動作が不安定で、どのように対処すればよいかわからないという問題に直面したRPAデベロッパーに、ワークフロー動作をより安定化するための実践的なテクニックを提供します。



本資料では上記のような課題を持っているRPAデベロッパーへ実践的なテクニックを提供する。

- <推奨のアクティビティ>**
- 要素を探す (Find Element)
 - 要素の有無を検出 (Element Exists)
 - 要素が出現したとき (On Element Appear)
 - 要素の消滅を待つ (Wait Element Vanish)
 - リトライスコープ (Retry Scope)
 - ⋮

左記のような便利なアクティビティをどのようなケースにどういう優先度で適用すればいいのかがわかる。



【困ること】
動作が不安定な時がある

目的要素のセレクター認識が
常に安定的に (※1) できている？

(※1) 安定的とは、想定されたセレクター要素を常に取得でき、目的動作が成功できること。

YES

NO

1. セレクター認識は安定的にできている

2. セレクター認識は安定的にできない

<解決方法> ※2

<解決方法> ※2

1-1. 画面描画前にロボットが動いてしまっている (ロボット処理が早すぎる、アプリが遅い、ネットワークが遅い)？

YES

1. プロパティ (Wait for Ready)
2. On Element Appear
3. Delay

NO

1-2. 要素が有効になる前に動いてしまっている (少し遅れてボタンが押せる)？

YES

1. プロパティ (Wait for Ready)
2. Find Element+Wait Attribute
3. Delay

NO

1-3. (無駄な) ある要素がランダムで表示される？

YES

1. プロパティ (Continue on Error)
2. Element Exist
3. Parallel

NO

1-4. ある処理の終了を適切に待てない？

YES

1. Wait Element Vanish
2. Delay

NO

1-5. アプリ起動に失敗する？

YES

1. Retry Scope

NO

実はセレクター認識は安定的にできていないことが原因では？ (右側へ)

2-1. セレクター認識に不安がある (たまに変わる)？

YES

1. セレクター修正 (ワイルドカード利用や nav調整)
2. Anchor Base
3. セレクターで変数を使う
4. Find Element + Find Relative
5. Get Attribute

NO

2-2. セレクター認識ができなかったため画像認識を使っているがたまにエラーが出てしまう？

YES

1. Send Hotkey
2. OCR
3. 画像認識のプロパティ (Accuracy)
4. AI Computer Vision

NO

有識者と一緒に高度な操作 (API利用、データベースアクセス等) を検討する。
あるいは業務での運用カバーを検討する。

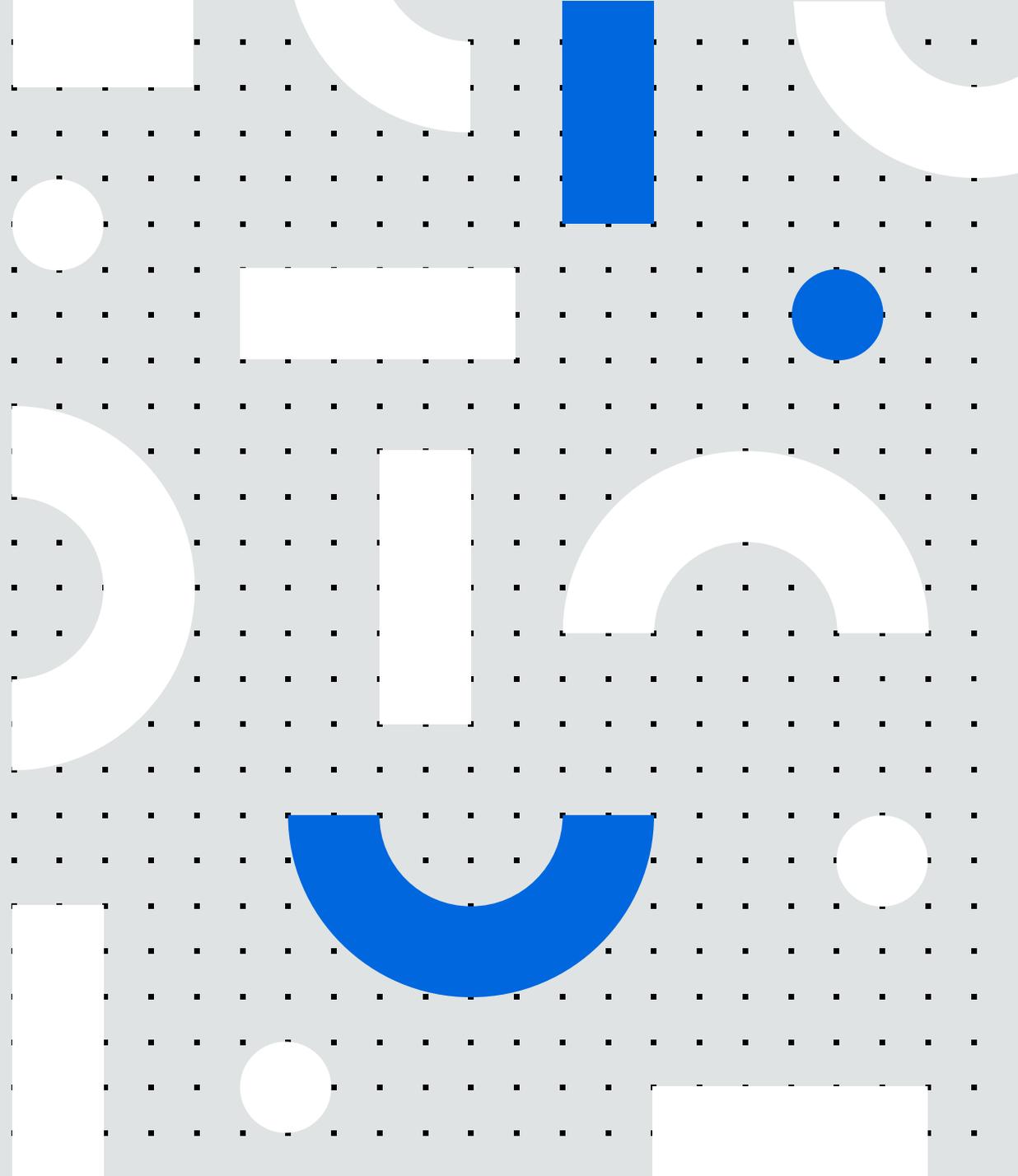
(※2) 各解決方法の技術的な説明はAppendix章をご参照ください。

本資料で検証用に使用したアクティビティのバージョンは下記となります。

UiPath.System.Activities 19.10.1

UiPath.UIAutomation.Activities 19.11.1

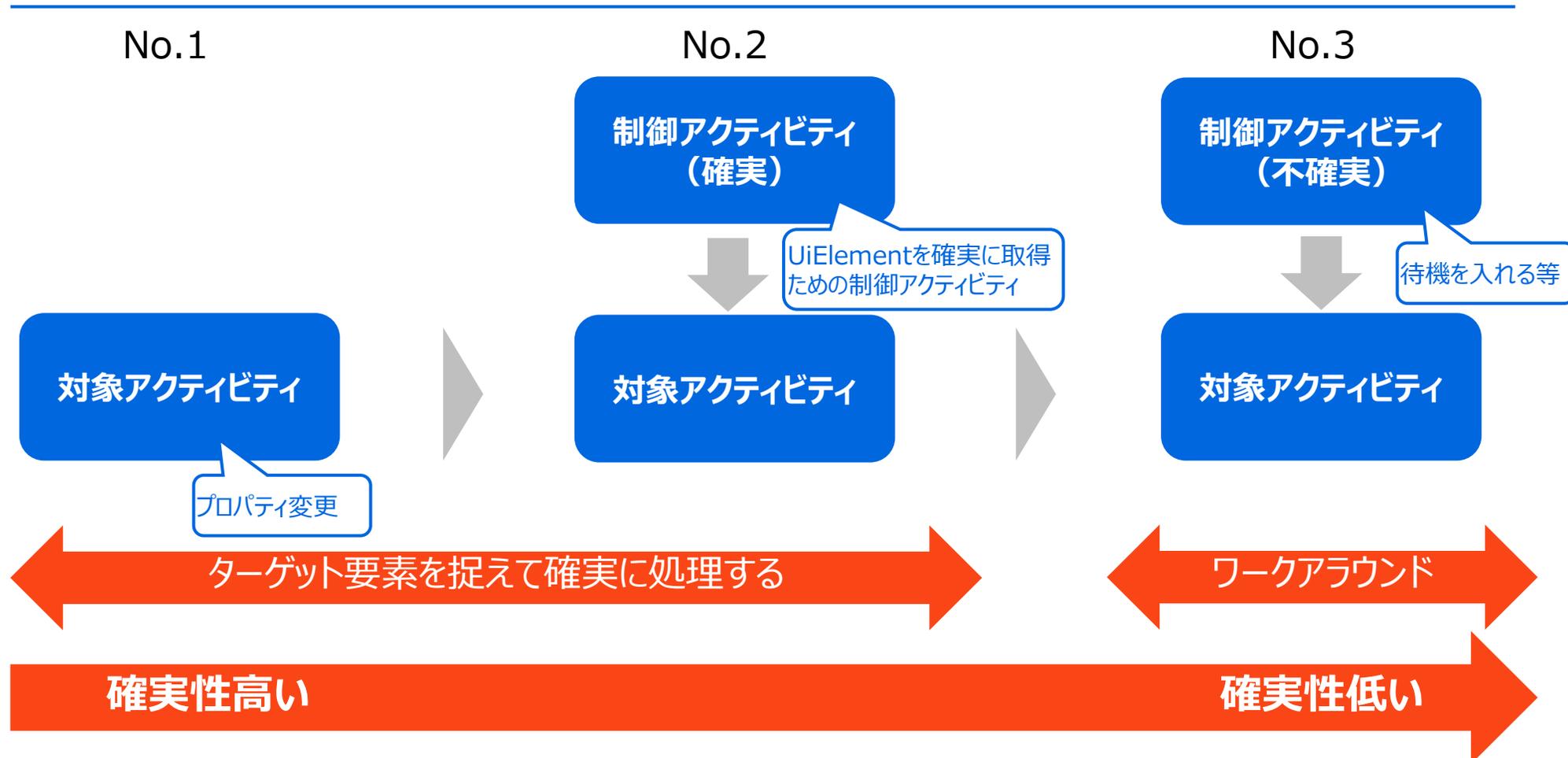
1. セレクター認識が安定的にできているが動作が不安定な場合



1-0. セレクター認識が安定的にできている時の解決方法優先度の考え方

各解決方法では、要素を捉えて確実に処理する方法から、不確実性のあるアクティビティを利用するワークアラウンド方法まで順に優先度を付けています。

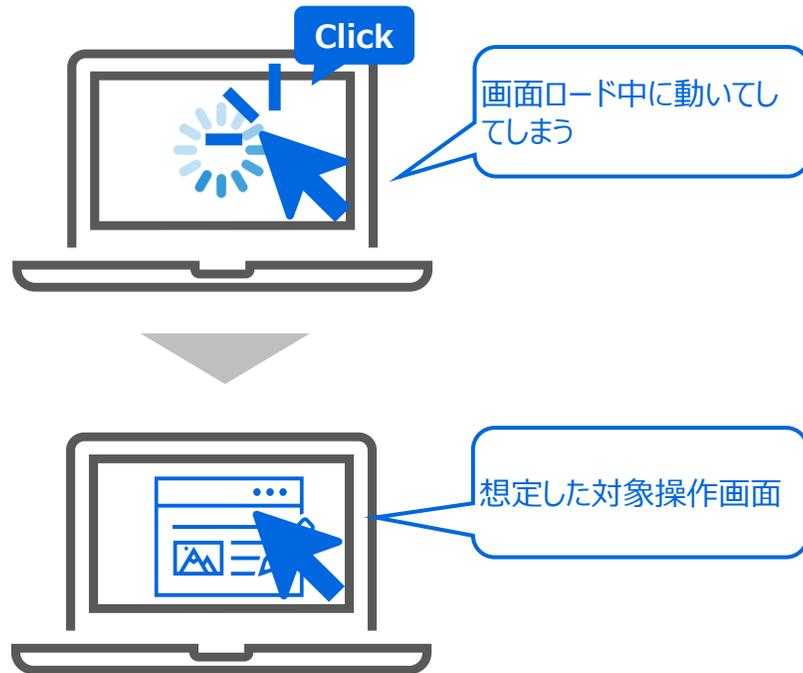
<解決方法の優先度>



1-1. 画面描画前にロボットが動いてしまっている (ロボット処理が早すぎる、アプリが遅い、ネットワークが遅い)

事象

ロボット処理が早すぎる、アプリが遅い、ネットワークが遅い等の理由で、画面描画前にロボットが動いてしまうケース



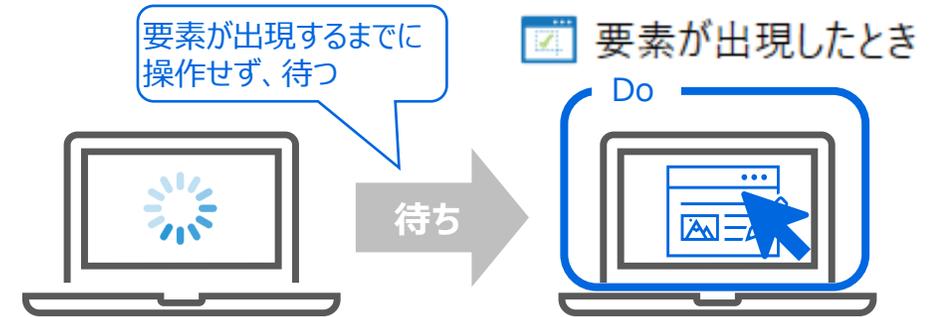
解決方法

No.1 プロパティ Wait for Ready (準備完了まで待機)

準備完了まで待機	対象の準備が完...
要素	NONE
テキスト	INTERACTIVE
共通	COMPLETE

アクティビティの準備完了まで待機を「COMPLETE」に設定する

No.2 On Element Appear (要素が出現したとき)



No.3 Delay (待機)



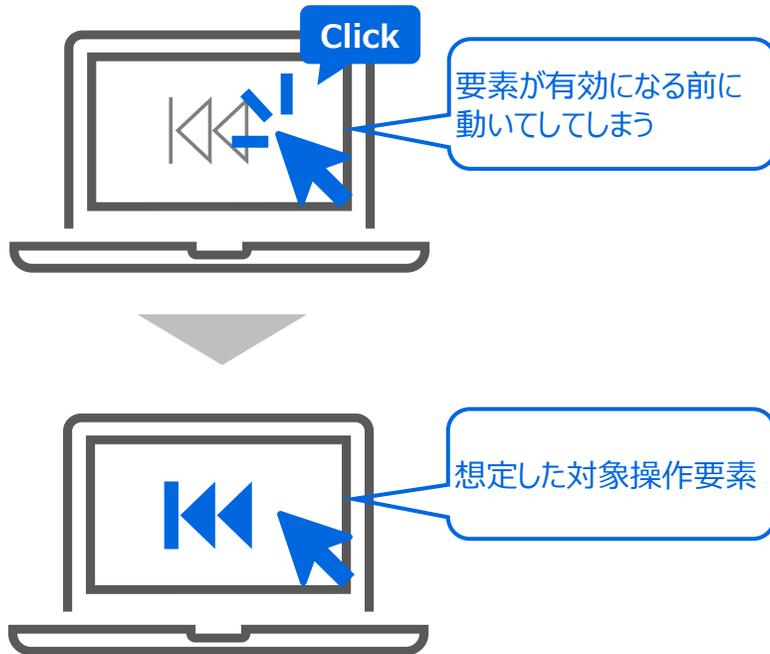
待機

ユーザーが手動操作時、対象操作画面が操作可能な状態までかかる時間を待機

1-2. 要素が有効になる前に動いてしまっている (少し遅れてボタンが押せる)

事象

要素が有効になる前にアクションしてしまうケース



解決方法

No.1 プロパティ Wait for Ready (準備完了まで待機)

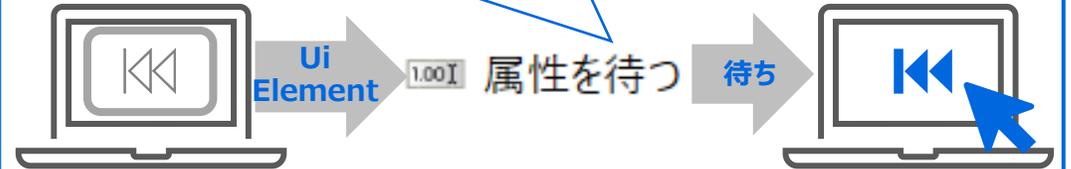
準備完了まで待機	▼ 対象の準備が完 ...
要素	NONE
テキスト	INTERACTIVE
共通	COMPLETE

アクティビティの準備完了まで待機を「COMPLETE」に設定する

No.2 Find Element+Wait Attribute(要素を探す+属性を待つ)

要素を探す

要素を探すで取得したUiElementを属性を待つに渡して、属性 (Visible、Enable等) が操作可能な状態になるまで待つ



No.3 Delay (待機)

🕒 待機

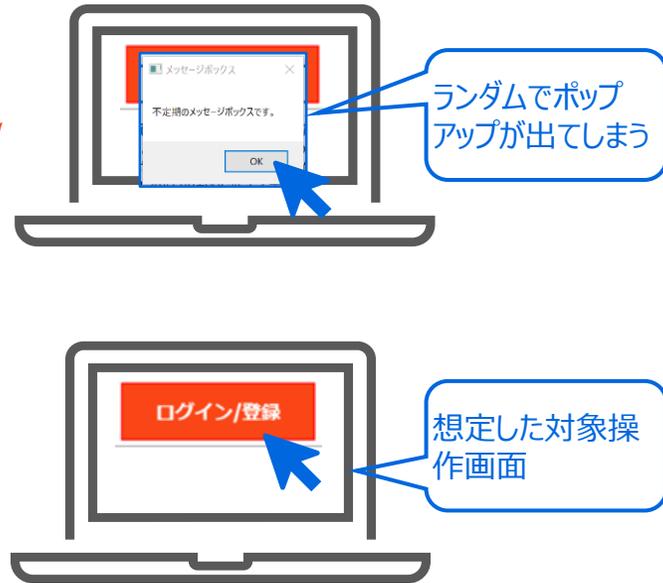
ユーザーが手動操作時、対象要素がアクティブになり、操作可能な状態までかかる時間を待機

1-3. (無駄な) ある要素がランダムで表示される

事象

(無駄な) ある要素がランダムで表示されるケース

前処理



解決方法

No.1 プロパティ Continue on Error(エラー発生時に実行を継続)

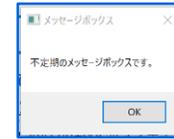
エラー発生時に実行を継続



True

ポップアップの「OK」ボタンを常にクリックする。クリックアクティビティのエラー発生時に実行を継続を「True」に設定

No.2 Element Exist (要素の有無を検出)



要素の有無を検出

True

存在する?

False

ポップアップ「OK」をクリック

「ログイン/登録」をクリック

No.3 Parallel (並列)

並列



並列でポップアップ「OK」クリック処理と正常処理を同時に実行させ、片方終わったら次に進む

1-4. ある処理の終了を適切に待てない

事象

何かの処理の終了を待っているケース

処理中...



プログレスバー消滅を待ち、継続の処理を行う必要がある



処理中のままファイルを開くとエラーになる

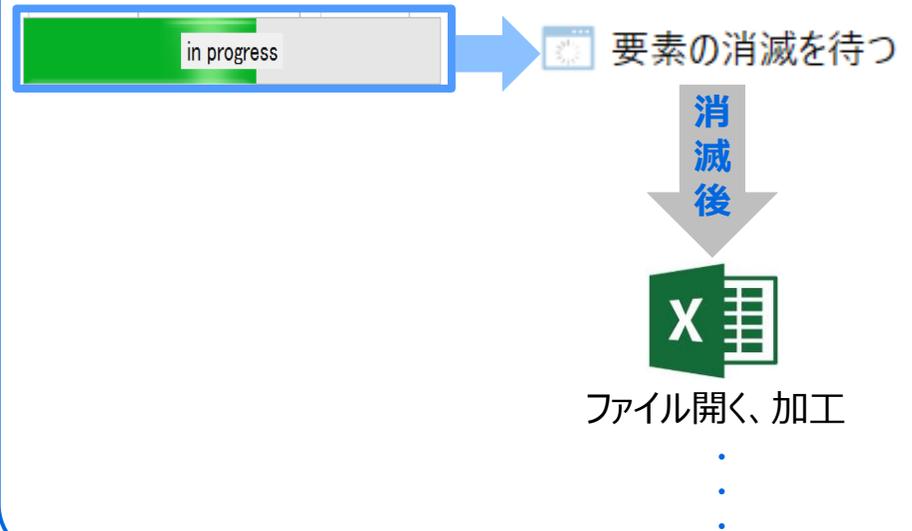
ファイル開く、加工



メール送信

解決方法

No.1 Wait Element Vanish (要素の消滅を待つ)



No.2 Delay (待機)

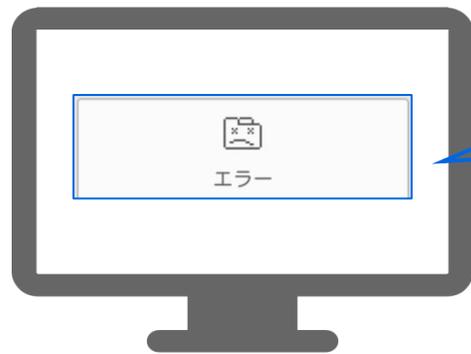


待機

処理が完了するまでかかる時間を待機

事象

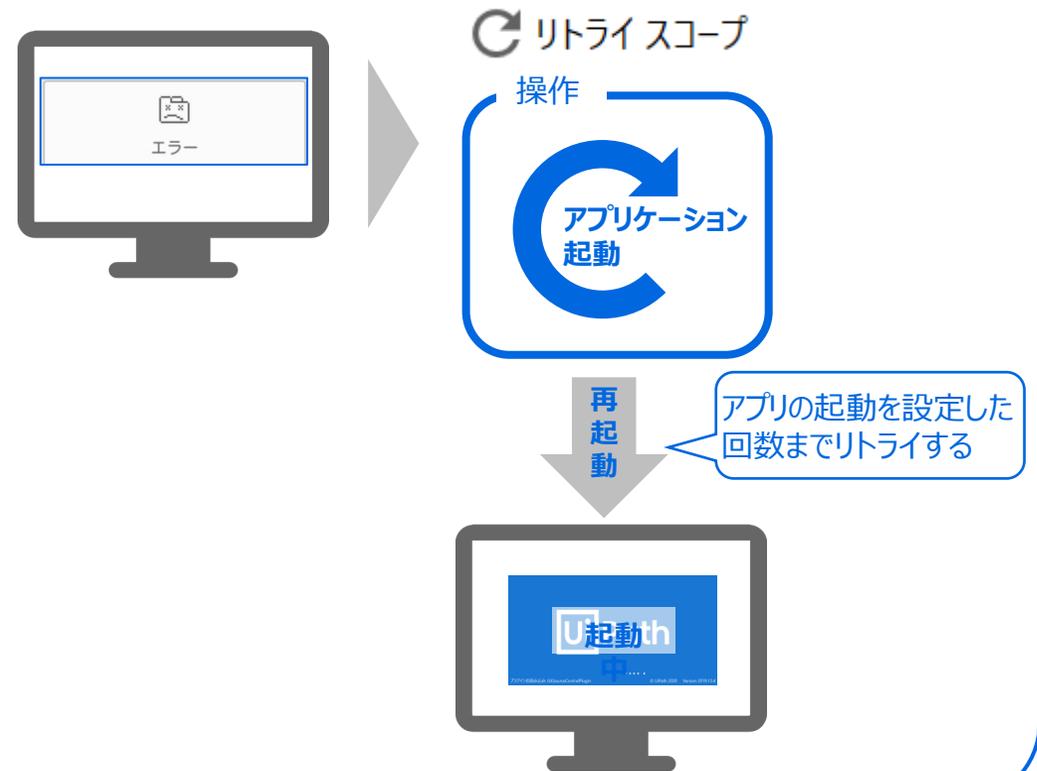
アプリ起動に失敗するケース



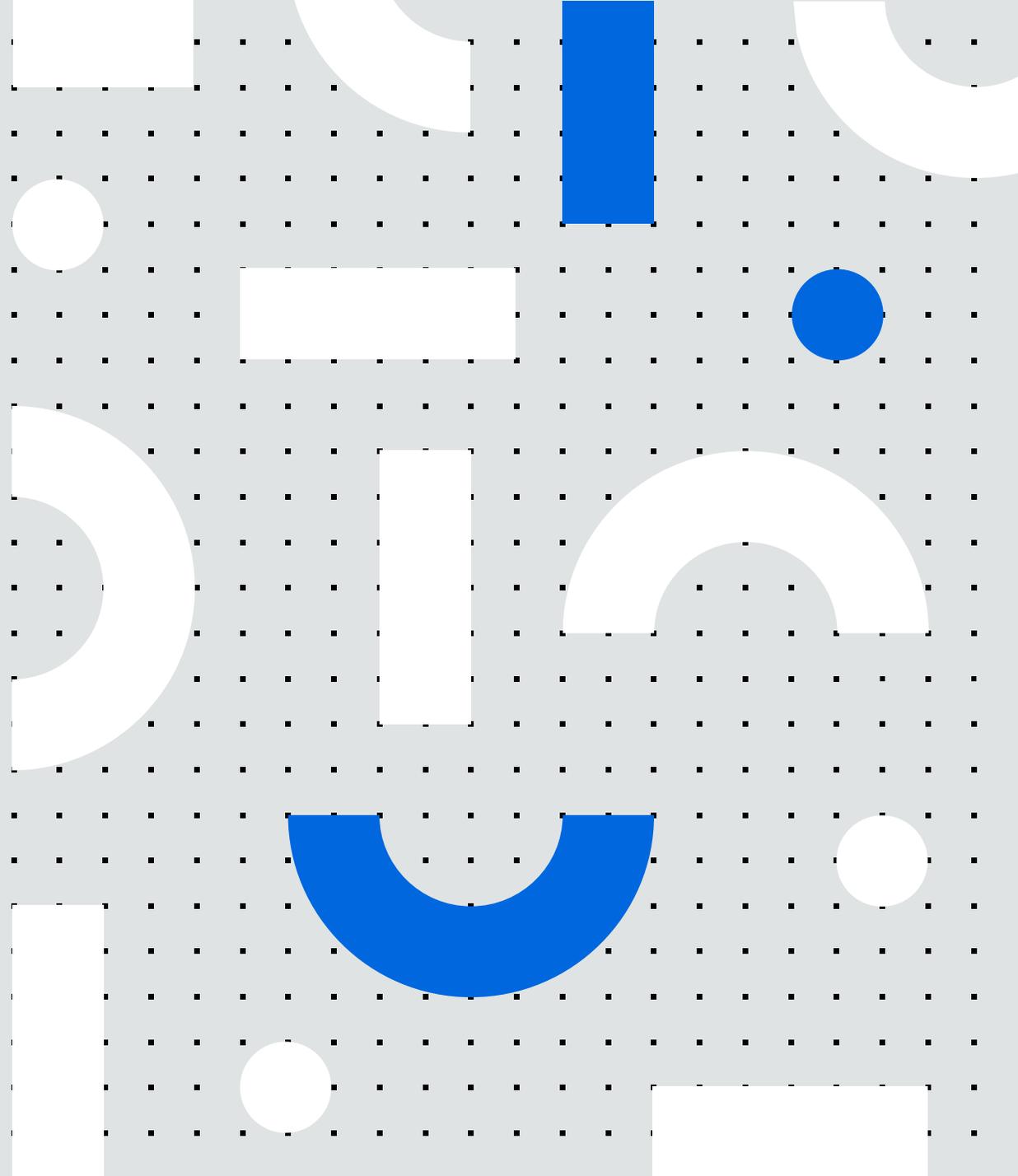
アプリ起動にたまにエラーになる

解決方法

No.1 Retry Scope (リトライスコープ)



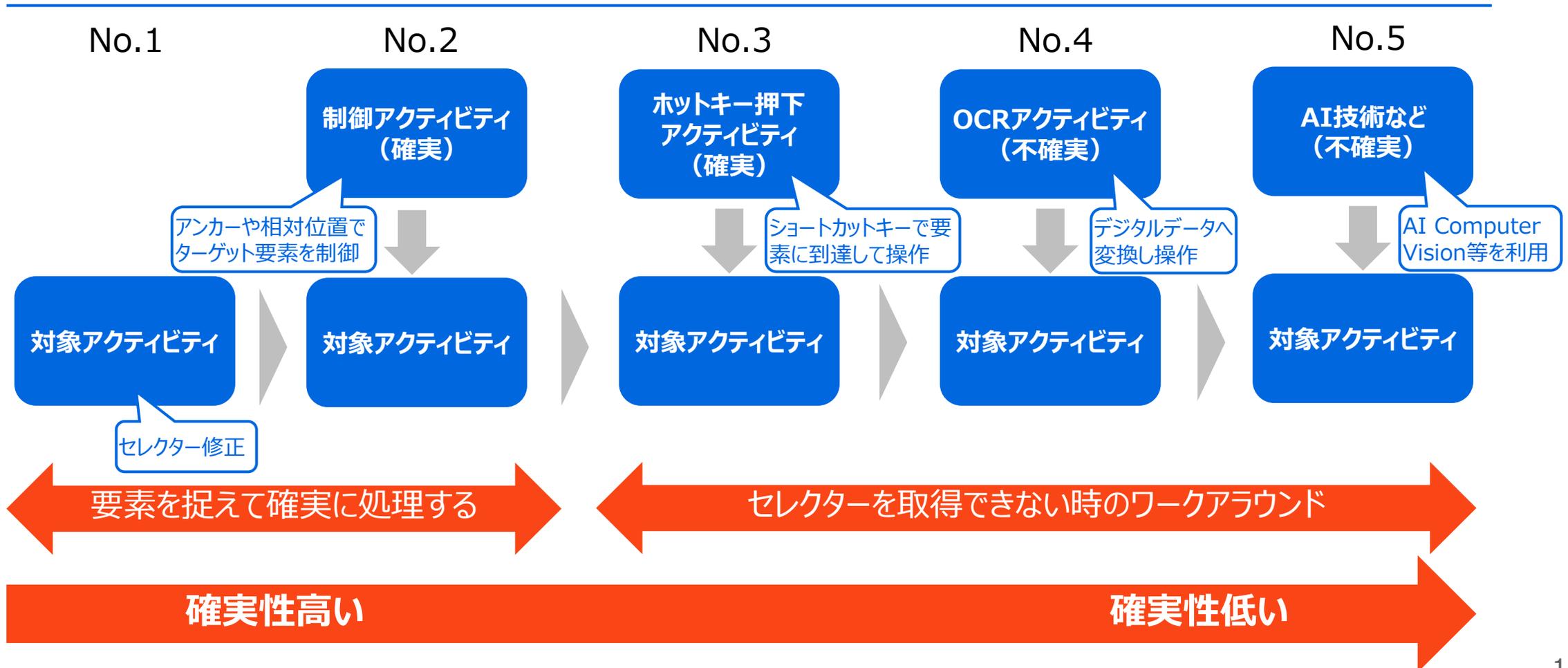
2. セレクター認識が安定的にできておらず動作が不安定な場合



2-0. セレクター認識が安定的にできていない時の解決方法優先度の考え方

各解決方法では、セレクターを修正して確実に処理する方法から、ショートカットキーや画像認識技術を工夫してワークアラウンド方法までの順に優先度を付けています。

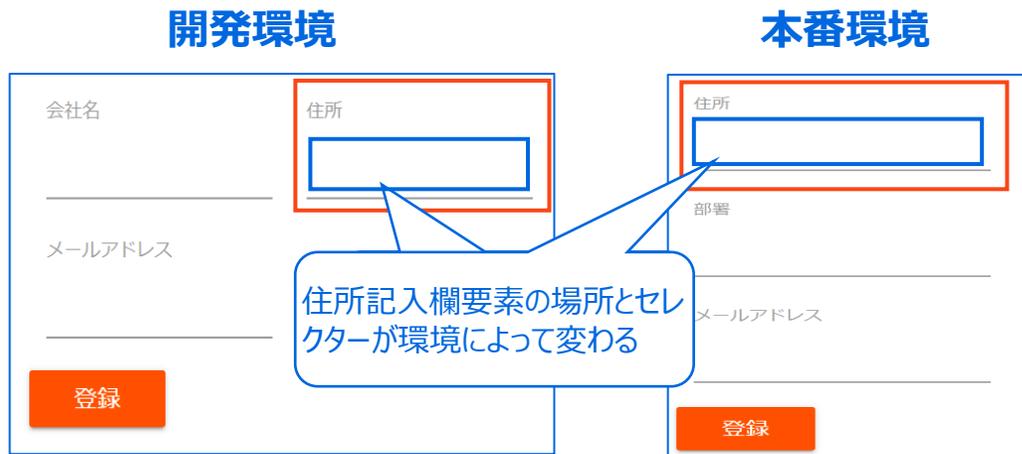
<解決方法の優先度>



2-1. セクター認識に不安がある (たまに変わる) (1/2)

事象

セクターが安定的ではない、たまに変わるケース①



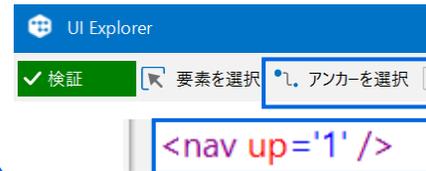
解決方法

No.1 セクター修正 (ワイルドカード利用やnav調整)
セクターにワイルドカードを利用

```
<html app='chrome.exe' title='Rpa Challenge 開発環境' />  
<webctrl innertext='住所' tag='INPUT' />  
  
<html app='chrome.exe' title='Rpa Challenge *' />  
<webctrl innertext='住所' tag='INPUT' />
```

環境によって異なる「title」に「*」を利用し修正

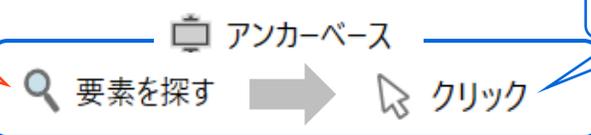
UI Explorerの「アンカーを選択」でnavタグを調整



UI Explorerで固定要素をアンカーとして選択し、要素同士の階層構造を利用して、目的の要素に到達

No.2 Anchor Base (アンカーベース)

セクター固定できる要素を探す

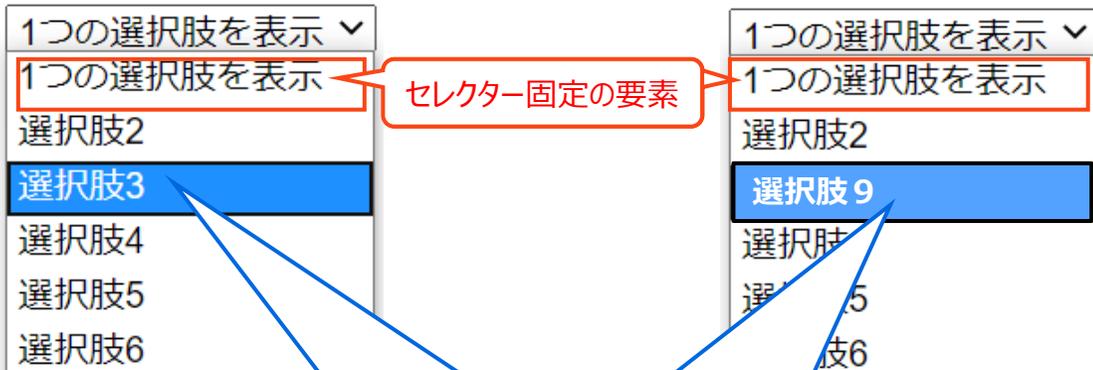


2-1. セクター認識に不安がある (たまに変わる) (2/2)

事象

セクターが安定的ではない、たまに変わるケース②

プルダウンを開く度に、セクターが変わる！！



目的要素 : いつもプルダウンメニューの3つ目を選びたいが、開く度に、セクターが変わる！！

解決方法

No.3 セクターで変数を使う

```
<webctrl name='example2' tag='SELECT' />  
<webctrl aaname='選択肢3' tag='OPTION' />  
  
<webctrl aaname='{{strName}}' tag='OPTION' />
```

変数で選択したい aanameを代入

No.4 FindElement+FindRelative(要素を探す+相対要素を探す)



No.5 Get Attribute (属性を取得)

```
<webctrl innertext='1つの選択肢を表示 選択肢2 選択肢3 選択肢4 選択肢5 選択肢6' />
```

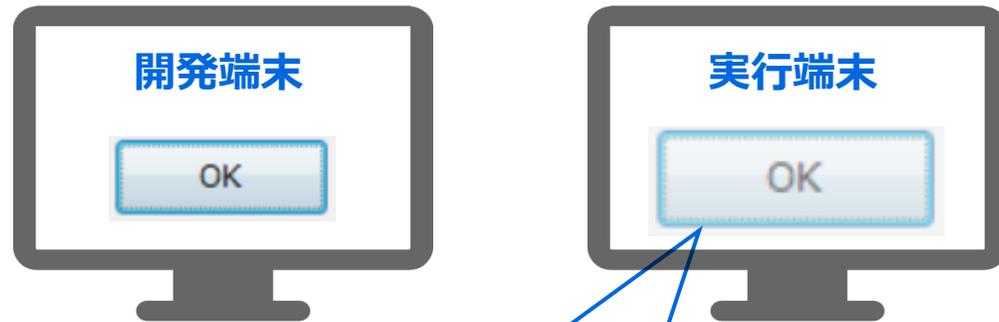
属性を取得

「innertext」属性の値をString型で取得し、**innertext.Split(" ").ToList(2).ToString**で3つ目の可変の値を取得し、aanameに代入

2-2. セレクター認識ができなかったため 画像認識を使っているがたまにエラーが出てしまう

事象

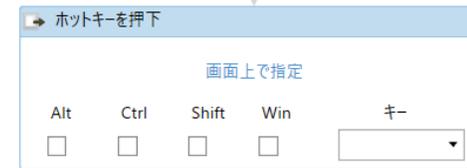
画像認識を使っているがたまにエラーが出てしまうケース



端末の解像度によって画像認識対象の画像がぼやけている

解決方法

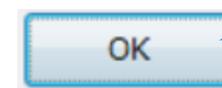
No.1 Send Hotkey (ホットキーを押下)



ショートカットキーで対象要素に到達して操作

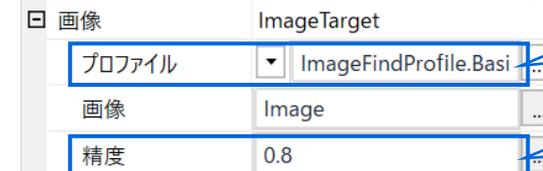
No.2 OCR

OCRでテキストを取得



OCRアクティビティでテキストを取得
→String型で取得("OK")

No.3 画像認識のプロパティAccuracy(精度)とProfile(プロファイル)



プロファイルをEnhancedに設定

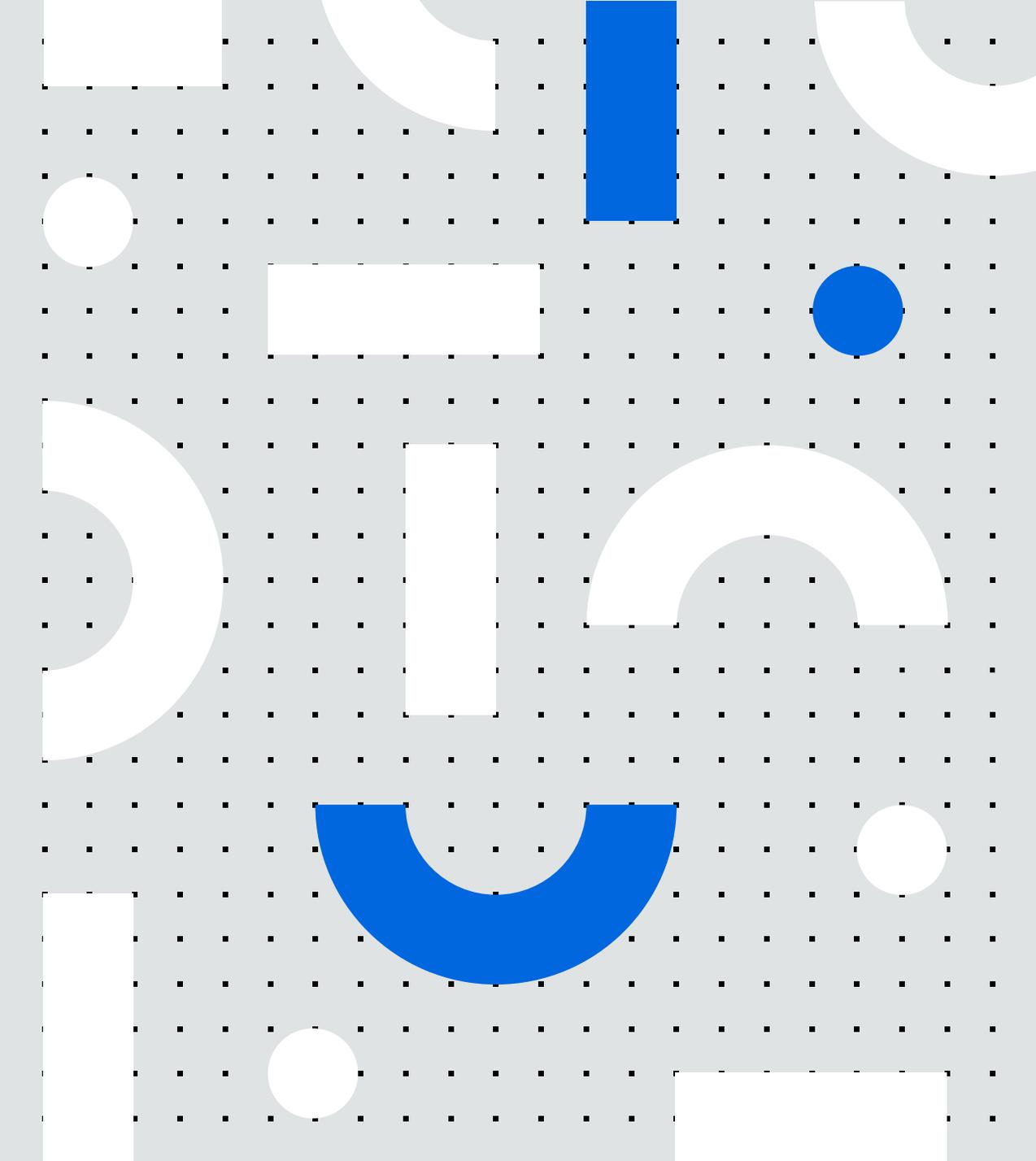
対象画像の精度を0~1で調整

No.4 AI Computer Vision



AI Computer Visionの技術を利用

3. Appendix



3-1. プロパティ Wait for Ready (準備完了まで待機) 詳細

説明

アクションを実行する前にターゲットが**レディ状態**になるまで待機します。

詳細

準備完了まで待機	対象の準備が完 ...	Web (IE)	デスクトップアプリケーション	SAP
要素	NONE	None	ターゲット UI 要素が見つかった場合を除き、何も待たない	
テキスト	INTERACTIVE	Interactive	対象UiElementが読み込まれるまで待つ	WindowsMessageのwm_nullの応答があるまで待つ(※1)
共通	COMPLETE	Complete	対象ページが全て読み込まれるまで待つ	専用のAPIを使用して実行状態になるまで待つ

(※1) WindowMessageを送信し応答があれば実行をします。 対象アプリがハングアップ状態のときは待ちますので、アプリケーションが独自で待機状態（くるくる表示、スプラッシュ画面など）を表現をしてもUiPathからすれば実行可能と判断してしまいます。人間なら、アプリケーションの待機表示があればクリックをしないはずですが、ロボットは容赦なくクリックします。自動化対象の業務で待機状態があるときは、「On Element Appear」などを使ってタイミングを調整してみてください。

詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/click#section-input>

説明

要素が表示されまで待機することができます。数秒後など出現するまで何秒かかるかわからない要素を捕まえるのに適しています。**UiElement** 変数として返します。**Do**シーケンス内のアクティビティは要素が表示された場合のみ実行されます。

詳細

待機時間を超えた場合にエラーを発生させたくない、あるいは、待機時間を超えた場合にはDoシーケンス内のアクティビティは実行されたくない場合に使います。

Repeat Forever（無限に繰り返す）をTrue（デフォルト）にして、タイムアウト時間を設定して、エラー発生時に実行を継続（ContinueOnError）をTrueにすると、タイムアウト時間を超えたらDoシーケンス内のアクティビティは実行しない、エラー発生しなく次の処理を実行します。



詳細は下記のガイドページをご参照ください。

<https://docs.uipath.com/activities/lang-ja/docs/on-ui-element-appear>

3-2. On Element Appear (要素が出現したとき) 詳細 (2/2)

WF詳細

プロパティ

UiPath.Core.Activities.OnUiElementAppear

オプション

- アクティブ化を待つ
- 無限に繰り返す False
- 表示されるまで待つ

その他

- プライベート

入力

ターゲット Target

クリッピング領域	
セレクター	"<html app='chronr ...
タイムアウト (ミリ秒)	5000
準備完了まで待機	INTERACTIVE
要素	VB の式を入力してく...

共通

- エラー発生時に実行を... True
- 表示名

出力

検出した要素

要素が出現したとき 'client'

readme

Do

文字を入力 'INPUT'

Email

"email"

SecureString で文字を入力 'INPUT'

Password

クリック 'BUTTON'

無限に繰り返すをTrueに設定すると、タイムアウト時間内に処理を繰り返す。

<Tips>
エラー発生時に実行を継続をTrueに設定すると、例外が発生しても、次の処理に進む。
無限に繰り返すをTrueに設定した場合、スロー (Throw) と組み合わせて利用する。

3-3. Find Element + Wait Attribute (要素を探す + 属性を待つ) 詳細 (1/2)

説明

要素を探すアクティビティは、指定した UI 要素が画面に表示される (フォアグラウンドに表示される) のを待って、**UiElement 変数として返します**。該当する**要素が出た時点ですぐ次のアクティビティに動作が移る**ため、固定待機に比べて実行時間が早くなります。

属性を待つアクティビティは、指定した UI 要素の属性の値が文字列と等しくなるまで待機します。

詳細

要素が有効になっているかどうかを確認する場合は有効です。取得したUiElement 変数を [属性を取得(Get Attribute)]または[属性を待機(Wait Attribute)]などのアクティビティに渡して、aastate 属性と組み合わせて使用します。



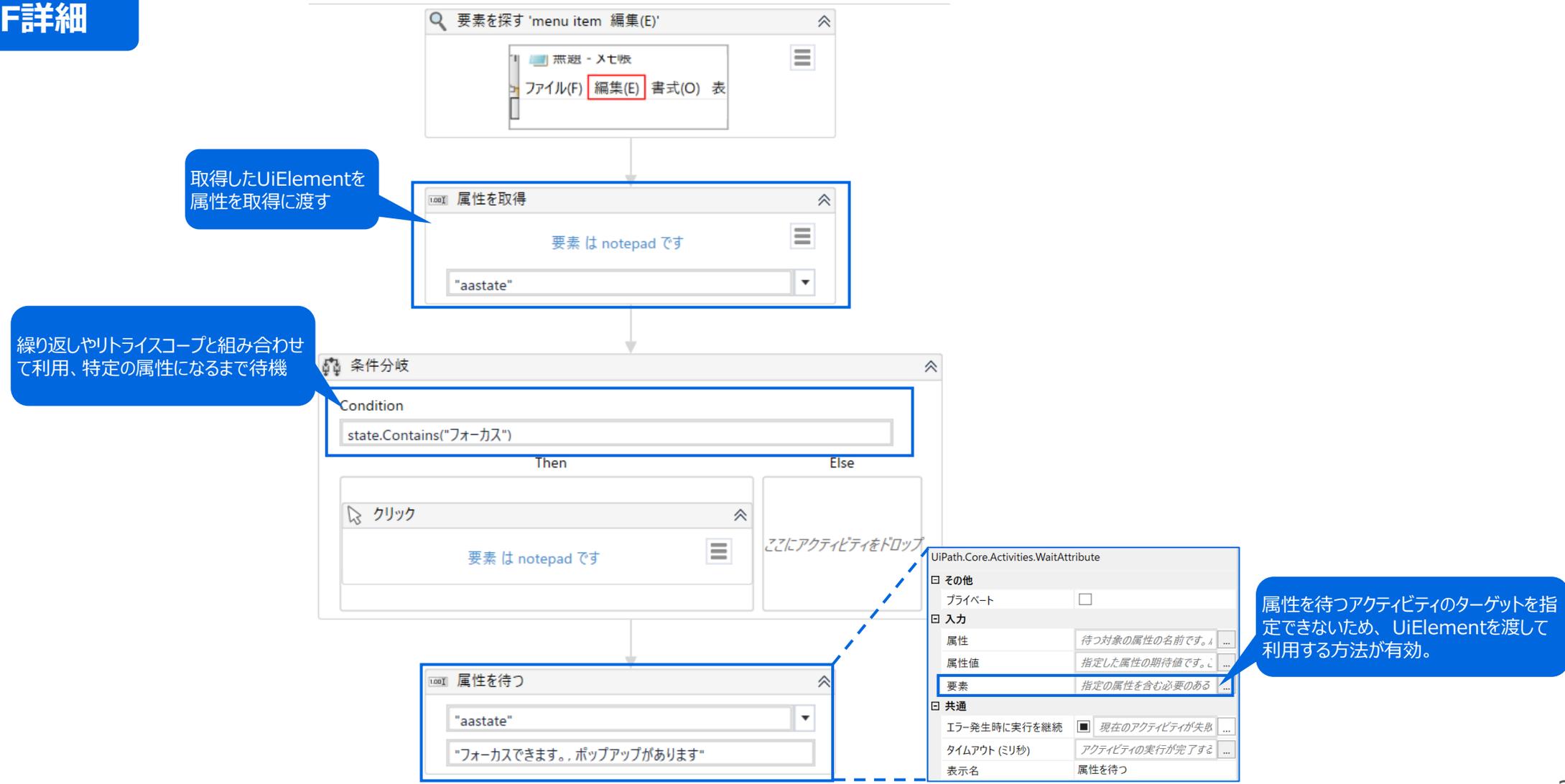
詳細は下記のガイドページをご参照ください。

<https://docs.uipath.com/activities/lang-ja/docs/wait-ui-element-appear>

<https://docs.uipath.com/activities/lang-ja/docs/wait-attribute>

3-3. Find Element + Wait Attribute (要素を探す + 属性を待つ) 詳細 (2/2)

WF詳細



3-4. Element Exists (要素の有無を検出) 詳細 (1/2)

説明

要素の有無を検出アクティビティではセクターで**指定した要素があるかどうかを判別**し、選択した要素があればTrue、なければFalseを返り値 (**Boolean型の変数**) として返します。

詳細

ある要素が出現したかどうかでその後の処理を変更したい場合に使用します。
例えば、メッセージがある場合にはウィンドウを閉じる等のアクションを入れることで、変則的にメッセージが発生する場合にUI要素が認識できずにエラーになってしまふ自体を防ぐことができます。

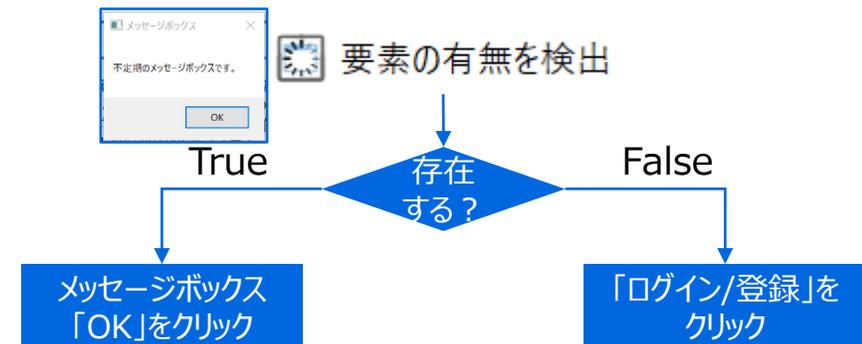
メッセージボックスが表示するかどうかによって処理が異なる



メッセージボックスが表示された場合「OK」ボタンをクリック



メッセージボックスが表示されない場合「ログイン/登録」をクリック



詳細は下記のガイドページをご参照ください。

<https://docs.uipath.com/activities/lang-ja/docs/ui-element-exists>

3-4. Element Exists (要素の有無を検出) 詳細 (2/2)

WF詳細

別途用意したタイムアウトの時間が経過するまで、要素の有無の判定を繰り返す処理をよく入れる。

存在しないかつ10秒以上である場合、次の処理に進む。

The workflow editor shows an 'Element Exists' activity. The 'Body' section contains a 'Loop' activity with a 'Repeat' type, which contains the 'Element Exists' activity. The 'Element Exists' activity has a 'Target' property set to 'Target', a 'Selector' set to '<webctrl aanam ...', and a 'Timeout (milliseconds)' set to '1000'. Below the loop is an 'A=B' activity with 'duration' set to 'Now-start'. The 'Condition' property is set to 'not exsit and duration.TotalSeconds < 10'. The 'Then' branch contains a 'Login Processing' activity with a 'Login/Registration' button. The 'Else' branch contains a 'Message Log' activity with 'Log Level' set to 'Info' and 'Message' set to '"ログインセッション残している"'.

<Tips>
タイムアウトの時間を短く設定することでより速く処理できる。

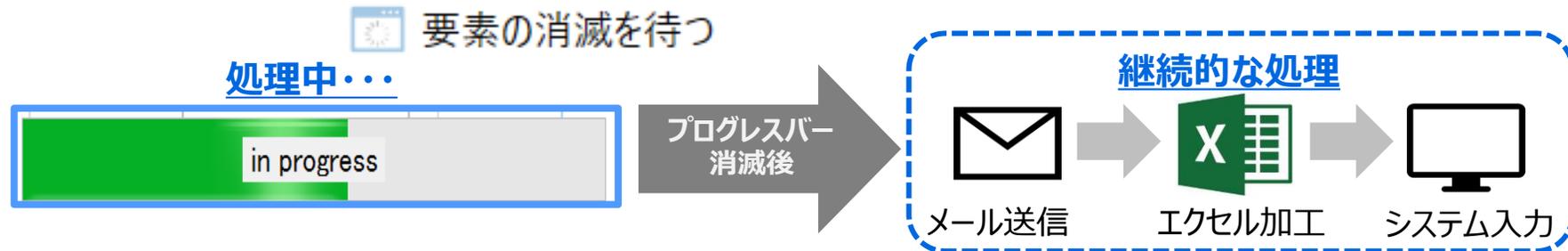
名前	変数の型
exsit	Boolean
start	DateTime
duration	TimeSpan

説明

指定した UI 要素が画面から表示されなくなるまで待機します。
呼び出されたときに「有」なら「無」になるまで待ち、呼び出されたときに「無」なら待たない、次の処理に行きます。

詳細

処理の進捗を表す要素がある、かつ該当処理が終了した後に、特定の画面要素がない場合に使います。
例えば、**印刷処理**、**ファイルのダウンロード処理**、**時間がかかる検索処理**（検索結果が不特定）、また終了した後に、転記やメール送信などの処理を行う場合に適用します。



詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/wait-ui-element-vanish>

3-5. Wait Element Vanish (要素の消滅を待つ) 詳細 (2/2)

WF詳細

実行中の処理が終了しましたら、次の処理を行う場合が適用します。

タイムアウト (デフォルト値で30秒) を適宜に調整しながら要素の消滅の待ち時間を設定する。

タイムアウトの時間が経つと、そしてエラー発生時に実行を継続 (デフォルト値でFalse) がTrueでなければ、タイムアウトにより例外が発生する。

3-6. Delay（待機）詳細（1/2）

説明

待機（Delay）アクティビティは、自動化処理をカスタム期間（hh:mm:ss形式）で**一時停止**します。

詳細

このアクティビティは、特定のアプリケーションが開始するのを待つ、または別の**アクティビティが実行完了するのを待つ**など、適切なタイミングを待つ必要とする際に使います。特に**要素を掴みにくい場合**を利用します。



詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/studio/lang-ja/docs/the-delay-activity>

3-6. Delay (待機) 詳細 (2/2)

WF詳細

Delay * X

Delay すべて展開 すべて折りたたみ

[↕] Delay

待機

プロパティ System.Activities.Statements.Delay

その他

プライベート

待機期間 00:00:00.1000000

共通

表示名	待機
-----	----

<Tips>
1秒以下の待機時間も指定できる。
<例>
0.1秒待ちの場合に「00:00:00.1」と書く。

説明

対象のアクションのあとに**想定通りの結果ができるまで対象のアクションを繰り返します**。
下記のいずれかを満たす間、内部に含めたアクティビティ群を所定の回数を実行します。
a) 条件 (Condition) に設定したアクティビティがFalseである
b) 内部に含めたアクティビティ群が例外をスローする

詳細

操作対象のシステムが不安定であり、アクション自体に失敗する時 (システム例外) がある場合に使います。
アクションのあとに想定通りの結果ができるまでに複数回の同様なアクションが必要である場合に使います。

システム不安定の場合



アプリケーションの起動チェック



詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/retry-scope>

<Tips>
Excelの場合、プロセスとして裏で動作する可能性があるため、Kill Process (プロセスの強制終了) アクティビティと合わせて利用を推奨する。

3-7. Retry Scope (リトライスコープ) 詳細 (2/2)

WF詳細

内部に含めたアクティビティ群が例外をスローする場合にリトライする。

条件 (Condition) に設定できるのはTrueかFalseのいずれを返す可能のアクティビティとなる。
要素の有無を検出 (Element Exists)
テキストの有無を確認 (Text Exists)
OCRでテキストの有無を確認 (OCR Text Exists)
画像の有無を確認 (Image Exists)
...
また、Workflow Manager Activities パッケージをインストールして、多数のBooleanで返すアクティビティも条件で利用できる。

3-8. Find Element + Find Relative (要素を探す + 相対要素を探す) 詳細 (1/2)

説明

要素を探すアクティビティは、指定したUI要素が画面に表示される (フォアグラウンドに表示される) のを待って、**UiElement変数として返します**。

相対要素を探すアクティビティは、固定要素を基準とする相対位置を使用してUI要素を検索し、**目的UiElementを取得できます**。

詳細

特定の位置にある要素を指定したい、かつ該当要素のセレクターが**可変**である場合は有効です。取得したUiElement変数を[相対要素を探す (Find Relative Element)]に渡して、オフセットを指定することで、特定位置の要素のUiElementを取得できます。



詳細は下記のガイドページをご参照ください。

<https://docs.uipath.com/activities/lang-ja/docs/wait-ui-element-appear>

<https://docs.uipath.com/activities/lang-ja/docs/find-relative>

3-8. Find Element + Find Relative (要素を探す + 相対要素を探す) 詳細 (2/2)

WF詳細

Workflow steps:

- クリック 'menu item 書式(O)'
- 要素を探す 'menu item 書式(O)'
- ハイライト (要素は uifont です)
- 相対要素を探す 'menu item フォント(F)...'
- ハイライト (要素は uirelative です)
- クリック (要素は uirelative です)

Properties window (UiPath.Core.Activities.FindRelative):

- その他: プライベート
- 入力: カーソル位置: CursorPosition
- Xのオフセット: 0
- Yのオフセット: 20
- 位置: Center
- ターゲット: Target
- クリッピング領域: []
- セレクター: VB の式を入力して...
- タイムアウト (ミリ秒): VB の式を入力して...
- 準備完了まで待機: [] 対象の準備が完...
- 要素: uifont
- 共通: エラー発生時に実行を... 現在のアクティビ...
- 表示名: 相対要素を探す 'menu i...
- 出力: 相対要素: uirelative

要素を探すで取得した固定要素が相対要素を探すのターゲット要素となる。

相対要素を探すに設定するオフセットは画面上に要素を指定やUI Explorerで「要素を選択」でカーソルの位置を確認できる。

UI Explorer elements:

- 広瀬アリス
- 清野菜名
- BENI
- ジェニファー・コネリー
- ファーストサマーウイカ

Cursor position: 858x378

Coordinates: 左x右: 608x1108, 上x下: 351x411

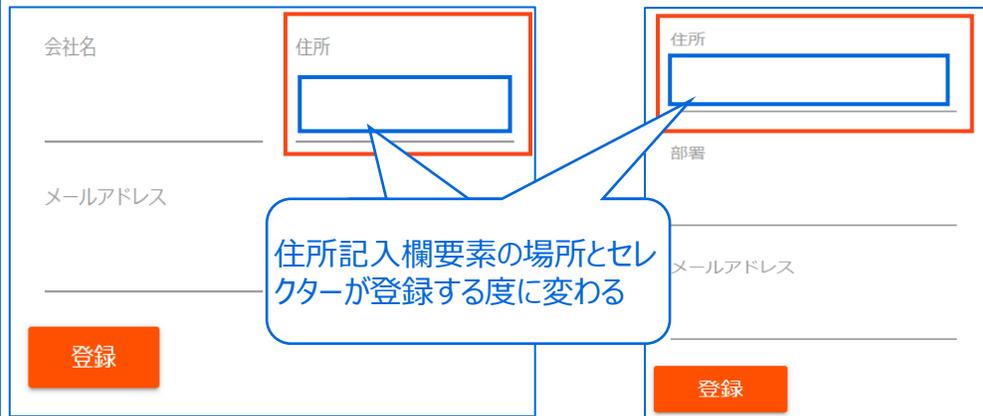
Actions: (ESC) - 選択をキャンセル, (F2) - 時間差で選択, (F3) - 領域を選択, (F4) - UIフレームワーク: 既定値

説明

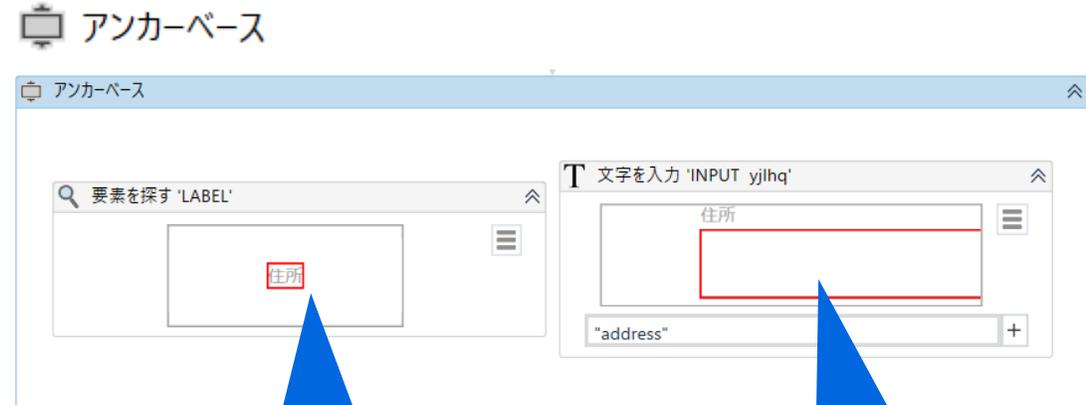
他のUI要素をアンカーとして使用してUI要素を検索するコンテナです。

詳細

参照する要素を指定すると、その要素との位置関係を利用して目的の要素に到達します。



詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/anchor-base>



参照する要素を探す。

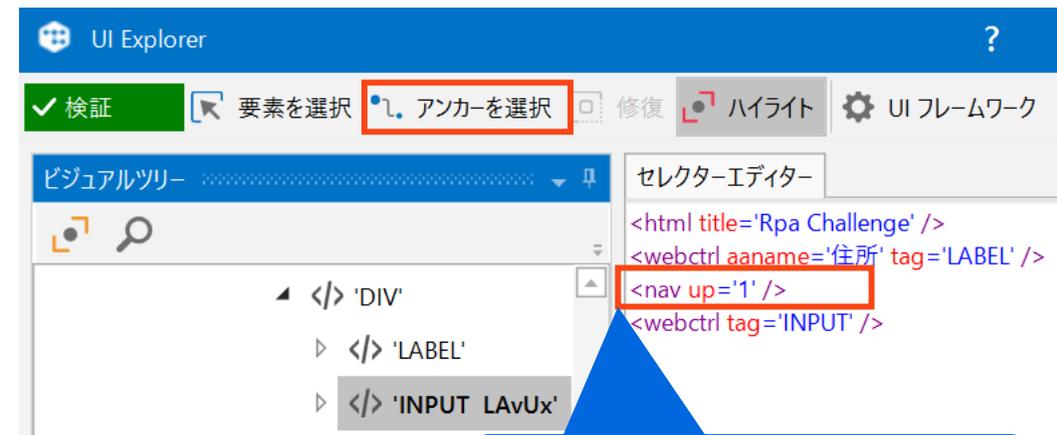
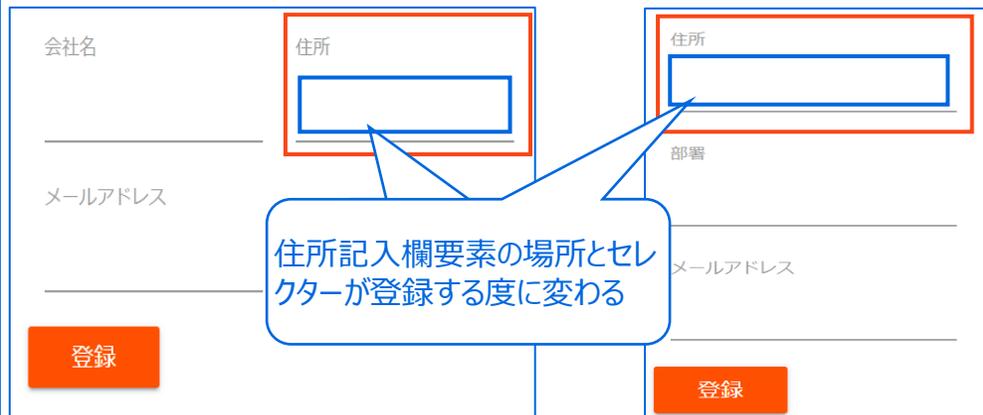
目的要素を操作する。

説明

UI Explorerでアンカーを利用します。
要素同士の階層構造を利用して、目的の要素に到達します。

詳細

要素同士の階層構造を利用して、UI Explorerの「アンカーを選択」を利用します。セクターが固定できる要素を選択して、**<nav>**タグの指定で目的の要素に到達します。



'LABEL'要素「住所」の1階層上の要素の配下にある'INPUT'タグのUI要素を指定する。

詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/studio/lang-ja/docs/selectors-with-wildcards>
<https://docs.uipath.com/studio/lang-ja/docs/about-selectors>

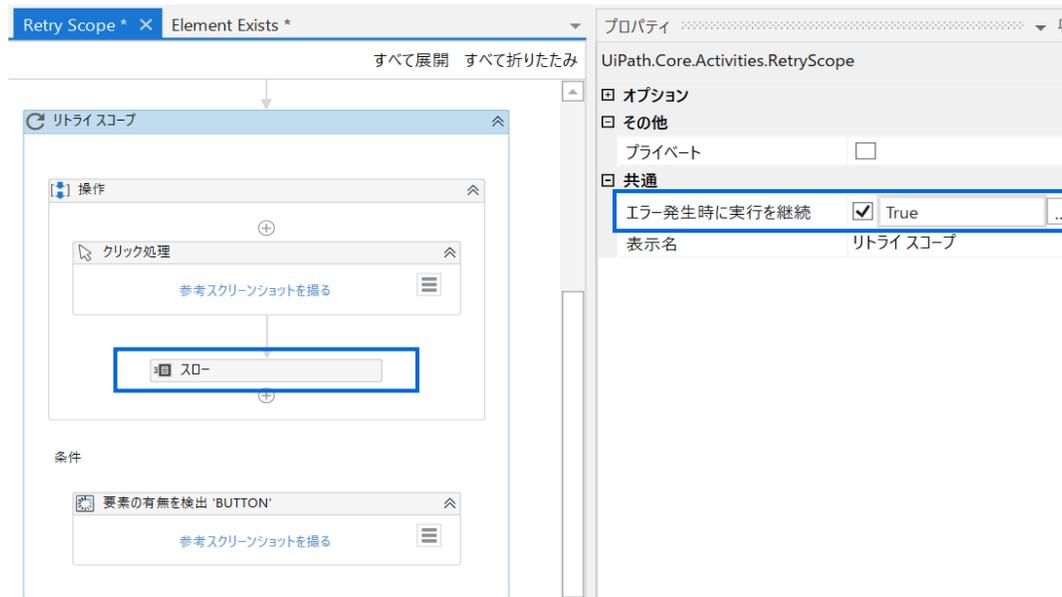
3-11. プロパティ Continue on Error (エラー発生時に実行を継続) 詳細

説明

該当するアクティビティの実行時に発生したすべてのエラーが無視されます。

詳細

予期せぬシステム例外でも、ビジネスロジックで組んだビジネス例外のスローでも、Trueに設定した場合に例外を無視して次の処理に実行します。



<注意>
「エラー発生時に実行を継続」を True に設定した場合、エラーが発生してもログに出力しないため、利用する際に、注釈や設計書に情報を残すことを推奨する。

詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/click#section-common>

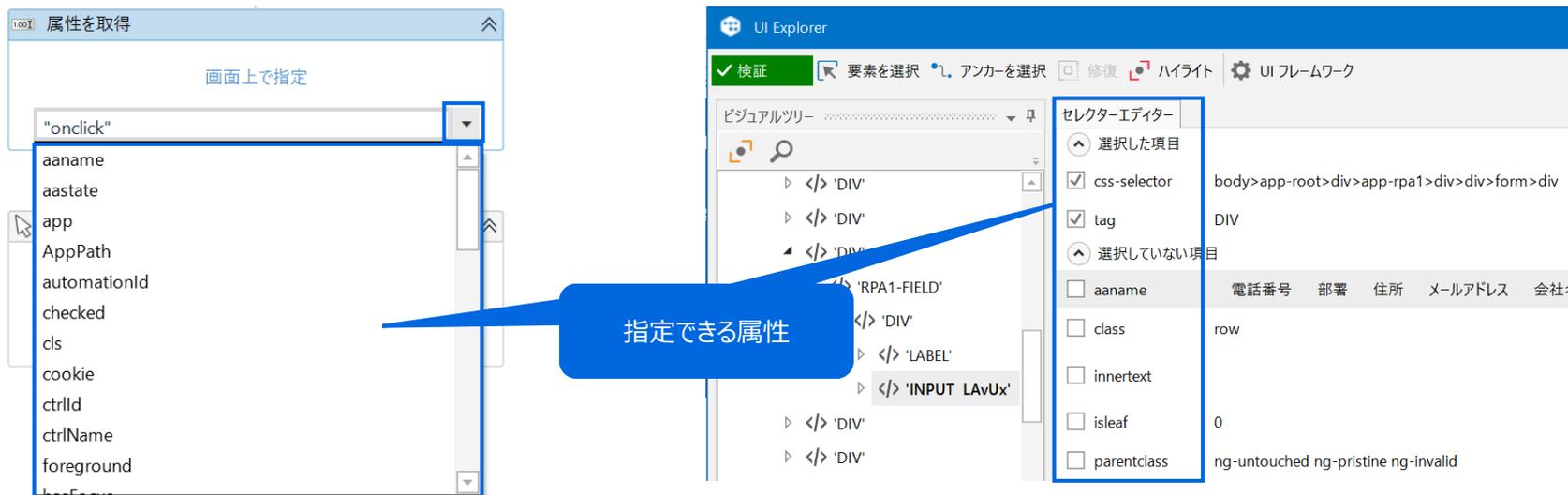
3-12. Get Attribute (属性を取得) 詳細

説明

UI 要素の指定した属性の値を取得します。要素が有効になっているかどうかを確認する場合は、「属性を待つ (Wait Attribute)」アクティビティと aastate 属性と組み合わせて使用してください。

詳細

UI Explorerのセレクトーエディターに表示されている属性を参照して、取得したい属性名を指定するのが有効です。



詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/get-attribute>

説明

セクターに変数を利用すると、変数の部分に文字列や数値を変動させることで、動的に画面上の要素を指定することができます。

詳細

カレンダー上の今日の日付など、画面上の要素を動的に指定したいときは、変数を使ってセクターを指定できます。

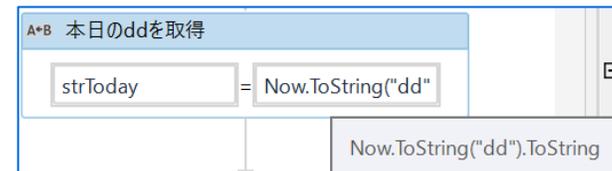
日曜日	月曜日	火曜日	水曜日	木曜日	金曜日	土曜日
1月31日	2月1日	2	3	4	5	6
7	8	9	10	11 建国記念の日	12 16:00 Block	13
14	15	16	17	18	19	20
21	22	23 天皇誕生日	24	25	26 16:00 Block	27

詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/studio/lang-ja/docs/dynamic-selectors>

セクターを編集

```
<html app='chrome.exe' title='予定表 - 2021年2月*' />  
<webctrl tag='DIV' innertext='10' />
```

'innertext'を日付ごとに可変に指定したい場合。



セクターを編集

```
<html app='chrome.exe' title='予定表 - 2021年2月*' />  
<webctrl tag='DIV' innertext='{{strToday}}' />
```

変数を代入する。

3-14. Send Hotkey (ホットキーを押下) 詳細

説明

UI 要素にキーボードショートカットを送信します。

詳細

特殊キーはキーリストから選択できます。

<注意>
「キー修飾子」と「ウィンドウメッセージを送信」と共に使用することはできない。

<Tips>
スペースキーを指定したい場合、半角スペース" "を打ち込む。

<Tips>
複数キーを同時に押下したい場合、「キー」にStringで記入する。

詳細は下記のガイドページをご参照ください。
<https://docs.uipath.com/activities/lang-ja/docs/send-hotkey>

3-15. 画像認識のプロパティAccuracy（精度）とProfile（プロファイル）詳細

説明

Accuracy（精度）とProfile（プロファイル）は画像を処理するアクティビティ（画像をクリック、画像を探す、画像が出現したときなど）のプロパティです。

詳細

Accuracy（精度）とProfile（プロファイル）の設定値を調整することで、画像認識の精確率を高めます。



Image.Accuracy - 検索しようとする画像と実際に見つける画像との間の最低限の類似度を表す 0 ~ 1 の測定単位である。見つける画像が検索しようとする画像とわずかに異なる場合に便利であり、既定値は 0.8 となる。

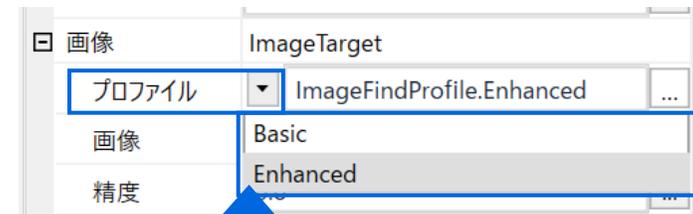


Image.Profile - 使用する画像検出アルゴリズムを変更することができる。Basic は、平均的な速度とリソースコストを持つ古典的なアルゴリズムである。Enhanced は、より多くのリソースを犠牲にして、より正確な結果を提供することができる。既定では、このフィールドはBasic に設定されている。

詳細は下記のガイドページをご参照ください。

<https://docs.uipath.com/activities/lang-ja/docs/wait-image-appear#section-input>

説明

OCR テクノロジーを使用するアクティビティは、端末の画面全体をスキャンして、表示されている文字を見つけ出してデジタルデータへ変換します。

詳細

各OCRアクティビティの詳細及び比較情報は下記のガイドページとブログをご参照ください。

➤ OCRのアクティビティ

<https://docs.uipath.com/studio/lang-ja/docs/ocr-activities>

➤ UiPath 組み込みの OCR アクティビティを利用するドキュメント処理プラットフォーム

<https://www.uipath.com/ja/blog/developer/ocr-activity-document>



他のOCRエンジンアクティビティへ変更可能となる。

説明

複数のアクティビティを内部のコンテナに含めて、並列で実行します。

詳細

並列 (Parallel) アクティビティの詳細解説は下記のブログをご参照ください。

➤ 並列 (Parallel) アクティビティについて

<https://www.uipath.com/ja/blog/developer/about-parallel-activities>

「条件」をTrueに設定した場合、含まれたアクティビティのうちひとつでも完了すれば次のアクティビティに遷移する。
「条件」をFalseに設定した場合、含まれたアクティビティのすべてが完了した後に次のアクティビティに遷移する。



説明

AI Computer Vision は、機械学習をベースとする手法により画面イメージから UI 要素を検出するとともにOCRエンジンによりテキストを抽出して、UIを完全に認識しようとします。

詳細

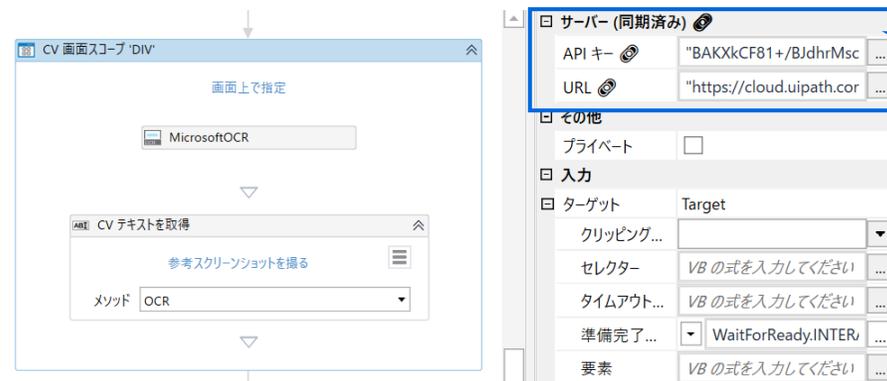
AI Computer Vision の詳細情報及び利用方法は下記の公式ページとアクティビティガイドページをご参照ください。

➤ AI Computer Vision リリースのお知らせ

<https://www.uipath.com/ja/resources/knowledge-base/ai-computer-vision-2019>

➤ AI Computer Visionのユーザーガイド

<https://docs.uipath.com/lang-ja/activities/docs/about-the-ui-automation-activities-pack#section-computer-vision>



「CV 画面スコープ」を利用し、サーバーとの同期情報が必要となる。
※取得・設定手順はユーザーガイドに参照。

Thank you



Legal Copy: Es quate in preperu mquaspitatia nobitiu menduntiam non nem facimen estrum faceaturenia voluptist, sum, voluptatiis et quodit aces del imi, quidell aborepello beat quatatus, sam im el et officiis estrum facepedio quae nonserum, ex et que doluptatius, que cuptureped quidi tentoresto moluptati odionecea nia quis ilit ma videbitesci ra poruntin cum con nost, sedis ex explab imporem. Hillore peria velis quae et mod quia sed ma volo temossitio que as sinci dem qui occulpa si ommolore susanda nihitis ellamet officiisit volupta temporumqui bero et quis undis eos elique pario. Et moluptae solore velicil itassi omnihiciis entempor re, voluptiunt occumquam, a verae pre debitat exerum voloriandi alit porehenihit, omnim eum cum venihit enihill enihillit, aut qui am dollam re duntio tene sunt.