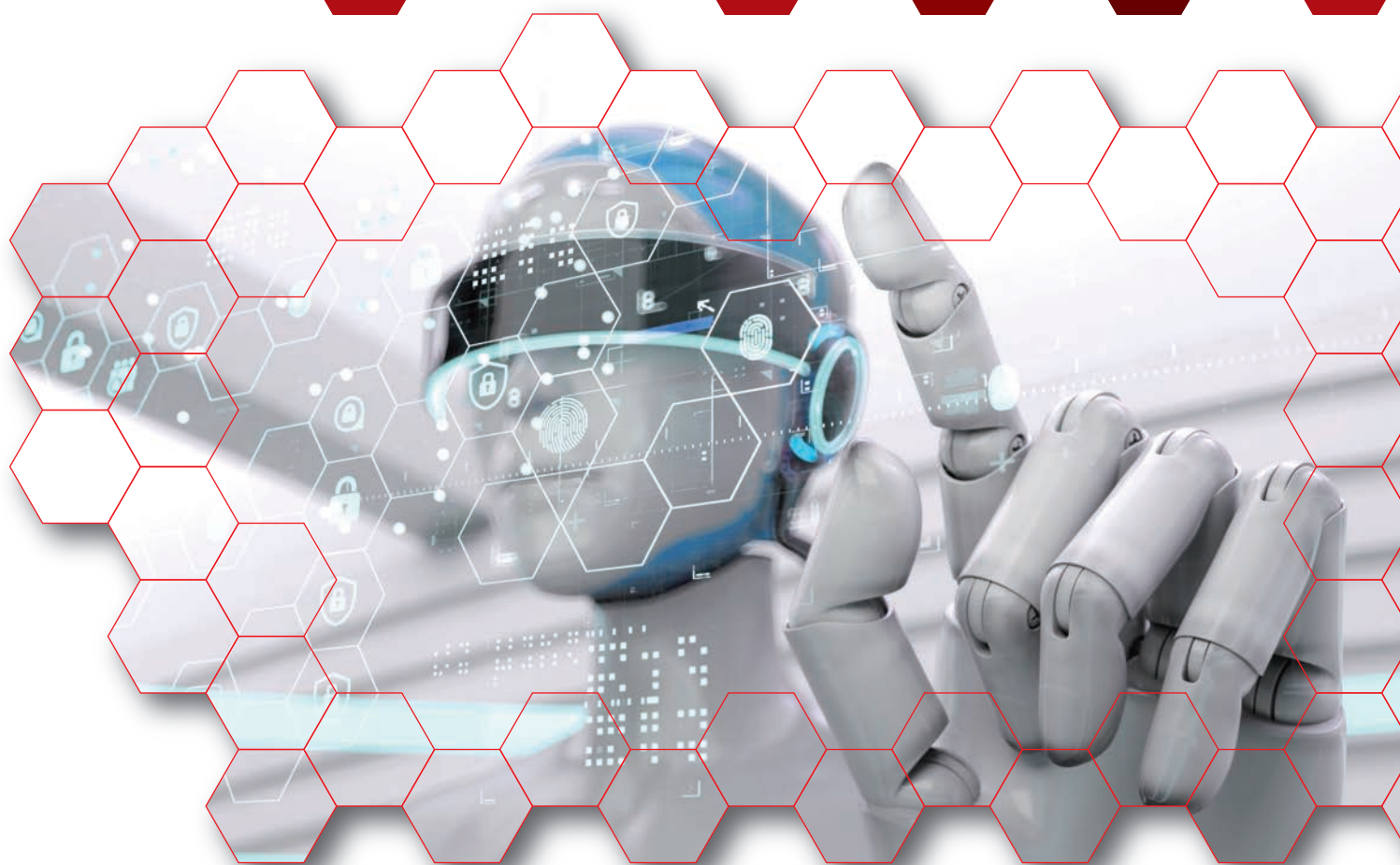




HeartCore Robo

RPA for ANYTHING



導入後継続率 **98%**

あらゆる業務アプリケーションの自動化を可能にする RPAソリューション



業務パッケージソフト

ERP・CRM・MS Office・FileMaker・CAD・AS/400 エミュレータ etc.



自社開発アプリケーション

Java・.NET・C・C++・C#・VB etc.



Webアプリケーション

IE・Chrome・Firefox・Safari・Edge etc.

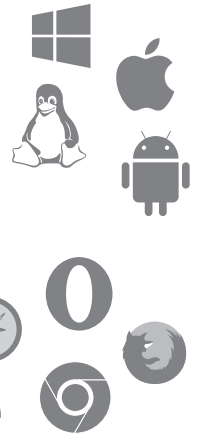


モバイルアプリケーション

iOS・Android

様々な環境で動作する RPAソリューション

- **マルチプラットフォーム**
(Windows / MacOS / Linux)
- **リモートデスクトップ・仮想環境**
- **全OSバージョンでの動作**
- **全ブラウザでの動作**



クイックスタートが可能

- 動作条件はJavaのみ
- 導入時のコストが必要最低限
(同時実行ライセンス形態)
- 実行環境へのモジュールインストール不要
- OSスケジューラを使用しての無人運用
(管理ポータル不要)

止まらない、安定した稼働

- スクリプト型シナリオを活用した
止まらないロボットの構築
- 高度なエラーハンドリング
- ピュアJavaコードの実行
- Javaプラグイン(カスタマイズ)の実装

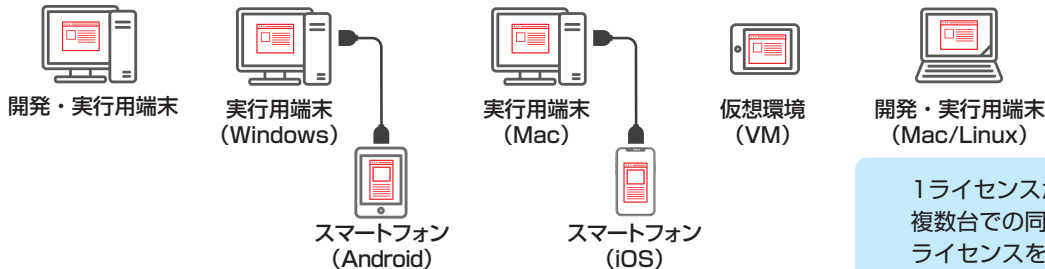
大規模なロボット運用が可能

- ロボットを監視する管理ポータル
- オーケストレーション機能
- シナリオの集中管理
- 実行履歴トレース

環境について

スモールスタートが可能

各端末のデスクトップ上で動作

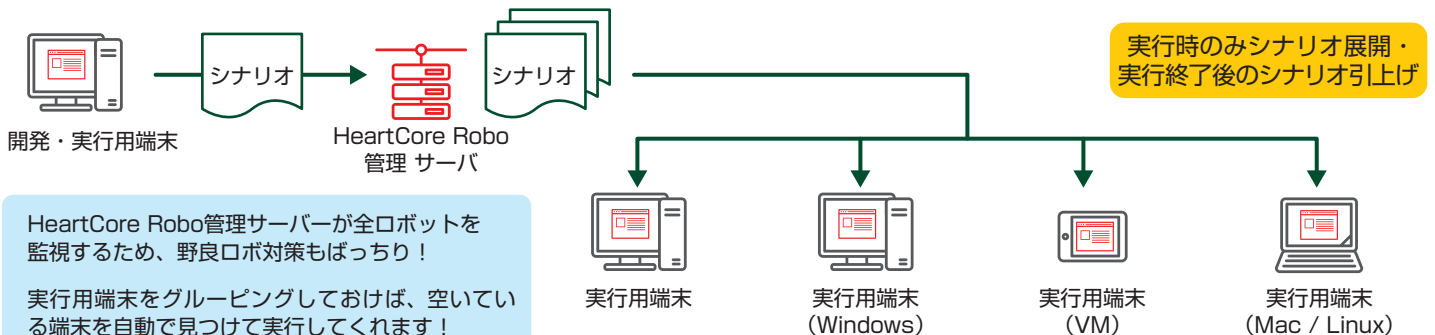


モジュールインストールは
何台でも可能

1ライセンスから始めていただけます！
複数台での同時実行が必要になった段階で、
ライセンスを追加すればOK！

大規模運用も安心

オーケストレーション機能によるロボット集中管理



実行時のみシナリオ展開・
実行終了後のシナリオ引上げ

HeartCore Robo管理サーバーが全ロボットを
監視するため、野良ロボ対策もばっちり！
実行用端末をグルーピングしておけば、空いている
端末を自動で見つけて実行してくれます！

コストパフォーマンス抜群のライセンス体系

1ライセンスで何台でもインストールが可能。実行ライセンスが消費されるのは実行時のみ。

- 実行ライセンス：シナリオを実行用端末で実行するためのライセンス
- 開発ライセンス：開発画面を開くのに必要なライセンス(実行ライセンスとしての役割も含む)

HeartCore Roboが**選ばれる理由**



高速イメージマッチング

- 独自の画像解析エンジンを用いて画面内の操作対象を高速判別(1秒以内)
- 操作アプリケーションのプラットフォームに依存しない高汎用性
- 色検知(例: 赤字のエラーメッセージ・請求書明細行の「空白」行)



ブラウザ自動操作

- Webページ内のHTML要素指定による自動操作
(例: ボタンのクリック・Webページ内フィールドへの入力)
- ページ内データ自動取得



OCR

- スキャン書類の読み込み→デジタルデータへの変換→業務アプリケーションへの入力
- 多様なOCRエンジン(Google Vision・Tesseract)
- フォント登録(例: ターミナル上に表示されるフォント)



バックエンドExcel操作

- 高速なExcelデータの読出・書込
- 実行端末へのExcelインストール不要



OSコマンド経由

- Windows / Unix系OSコマンドの活用
- 確実なファイル操作が可能(フォルダ内ファイル一覧取得・ファイル・フォルダの新規作成・コピー・移動など)

開発について

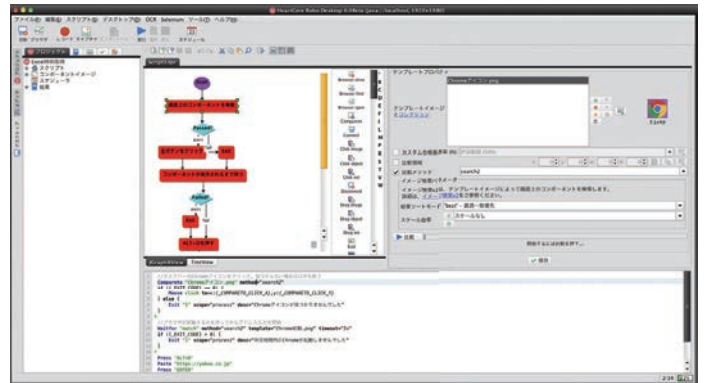
フローチャート型とスクリプト型の両方に対応

誰でもロボット構築ができる直感的なフローチャート型

- 各コマンドはアイコンから選択
- ドラッグ&ドロップで組み立て
- プロパティ画面で詳細な調整が可能
- グラフィカルビューとツリービューを搭載

精緻なロボット構築を可能にするスクリプト型

- シンプルな文法
- ウィザード入力補助によるコマンド挿入
- Javaの活用によりプロフェッショナルな要件にも対応
- シミュレータ活用による逐次動作確認



プロフェッショナルな要件に対応可能

- Javaコードの活用
- Javaプラグイン開発による独自メニューの追加
- モバイルアプリケーションテスト
- 画像差分比較・差分検知レポート

モバイルアプリケーションテスト



```

Javaブロック_コマンド付け.ipr
1 //コマンド付け
2 Eval Price="1000000"
3
4 Java {
5
6     import java.text.NumberFormat;
7
8     //String str_value = getContext().getVariable("Price").toString();
9     int int_value = getContext().getVariableAsInt("Price");
10
11     //NumberFormat インスタンスを生成
12     NumberFormat nf = NumberFormat.getInstance();
13
14     //getContext().setVariable("Price", nf.format(Integer.parseInt(str_value)));
15     getContext().setVariable("Price", nf.format(int_value));
16 }
17
18 Log [Price]
19 Report "results.xml"
    
```

HeartCore Roboのコードと
Javaコードのハイブリッド型開発

テンプレートイメージとの
比較・差分検知(カード移動部分)

5秒以内に検知



機能一覧 [開発]

機能	説明
多様な開発環境	<ul style="list-style-type: none"> ローカルデスクトップ VNC (Virtual Network Computing) / RDP (Remote Desktop Protocol) Xcode for iOS ADB for Androids オフライン環境 (静的イメージ使用)
日本語対応	<ul style="list-style-type: none"> メニュー・ヘルプ・チュートリアルすべて日本語対応
画面自動最適化	<ul style="list-style-type: none"> ローカルデスクトップ:解像度自動検出・複数モニターの使用が可能 VNC/RDP/iOS/Android:開発画面ビューア表示・自動スケールが可能
多様な検知メソッド	<ol style="list-style-type: none"> 高速イメージマッチング (画像・アイコン・ラベル・GUIコンポーネント) HTML解析 (ブラウザ自動化) 色マッチング (RGB指定 - 例: エラーメッセージ、GIS) OCR (非デジタル文字の検知) 文字画像コレクション比較 (OCRで検知出来ないフォントの検知) デスクトップ座標指定 (画面X、Y座標) API (Excel/CSV/テキストファイルの読み込み・書き込み)
マルチプラットフォーム	<ul style="list-style-type: none"> すべてのプラットフォーム
ロボット動作対象アプリケーション	<ul style="list-style-type: none"> イメージ認識が出来る全てのアプリケーション (Flash認識可能) アドオン・設定変更が不要
ルック&フィール変更	<ul style="list-style-type: none"> 10種類以上のテーマから選択 スクリプトエディタの表示変更が可能

機能	説明
直感的な開発 (スクリプト & フローチャート)	<ul style="list-style-type: none"> ドラッグ&ドロップによるフローチャート型開発、ウィザード入力補助によるスクリプト型開発の両方に対応 自然言語に近い簡素なスクリプト (例: "Wait Is", "Press CTRL +C", "Excel open..") 操作録画によるスクリプトの自動生成が可能 録画されたスクリプトは全て右クリック>プロパティで調整可能 マルチプラットフォームでのスクリプト実行可能 (例: Windows⇔Mac の転用) Java への変換が可能
スクリプトエディタ	<ul style="list-style-type: none"> フォーマット機能付スクリプトエディタ標準装備 自動シンタックスチェック オートセーブ機能
コードテンプレート	<ul style="list-style-type: none"> 使用頻度の高い処理のテンプレート (例: 対象画像が表示されるまで待つ・ページをスクロールダウンさせる・Excelファイルの読み書き) 50以上のテンプレート+独自テンプレートの追加が可能
高性能ロボット作成	<ul style="list-style-type: none"> 実行変数・If/For文使用による複雑な処理 100種類以上の多様なシステム変数の利用が可能 (例: 実行OS情報・現在のマウス位置 (x, y)、対象オブジェクトのRGB値) 高頻度処理のプロシージャ化 ロボット内複数スクリプトのスケジュール設定可能 ロボット端末のOSコマンド実行 (例: exeファイル) ロボット実行時に対象が認識出来ない場合の詳細なエラーハンドリング 平文で表示出来ない文字情報のマスキング (例: パスワード情報)

機能一覧 [検証]

機能	説明
ロボットシミュレーション	<ul style="list-style-type: none"> 対象物検知シミュレーションの実行が可能 (実行時と同一の検知エンジンを使用) 正確なイメージテンプレートの再定義を行う事による検知率の向上 (=ランタイム時の認識失敗リスクの低減) が可能 全ての検知メソッドによるシミュレーションが可能 (イメージ・色・OCR・文字) 検知に要する処理時間の把握が可能 ヒットした対象物全ての座標がメモリに格納される (例: 2,3,4..n番目にヒットした対象物をアンカーとすることが可能)
検知ロジックのチューニング	<ul style="list-style-type: none"> 一致率の加減 (10-100%の範囲) が可能 複数対象物がヒットする場合の結果ランキングソート (一致率・左上・最左・最右・最上部・最下部) 対象範囲を絞り込む事で検知速度の向上が可能
検知・非検知時の詳細設定	<p>ヒットした場合・ヒットしない場合両方のハンドリングが可能 例: ヒットした対象物をアンカーとして右横のフィールドにある値をコピーする</p> <p>例: ヒットしなかった場合には別のボタンをクリックする・ヒットするまで待つ</p>
デバッグ	<ul style="list-style-type: none"> スクリプトエディタ上でのブレークポイント設定 部分実行によるデバッグ エラー時のスクリーンショットとイメージテンプレートとの一致率データの取得が可能

機能一覧 [展開・実行・管理]

機能	説明
簡易な展開	<ul style="list-style-type: none"> 実行用HeartCore Roboライブラリはインストール不要 ロボットファイル (スクリプト・イメージテンプレートのみ) の配置でセットアップ完了 実行時のパラメータ指定を行うCLIコマンドの作成が可能
多様な起動方法	<ul style="list-style-type: none"> ユーザによるマニュアル起動 スケジュールによる起動 (CLI経由のサイレント起動が可能) HeartCore RAP (Robot Automation Portal:管理ポータル) からの実行
ロボットモニター	<ul style="list-style-type: none"> HeartCore RAP からリアルタイムのステータス確認が可能 HeartCore Robo開発環境からVNC/RDP経由で実働状況のモニターが可能
干渉防止	<ul style="list-style-type: none"> ユーザのマウス・キーボード操作による干渉を無効にする事が可能
アラート	<ul style="list-style-type: none"> 登録したメールアドレスに成功時とエラー時にはその内容の送信が可能 (メーラーの設定不要)
カスタマイズ可能なレポート	<ul style="list-style-type: none"> 任意のステップでのスクリーンショット (例: 「データ登録」ボタンをクリックする際の画面キャプチャ) 検知エラー時のスクリーンキャプチャと比較が可能
ポータル	<ul style="list-style-type: none"> 全ロボットの実行状況監視 管理下端末へのロボットの展開 ダッシュボードからの全ロボット動作実績把握 異なるロボットとの連携スケジュール設定が可能

機能一覧 [その他]

機能	説明
リモート保守	<ul style="list-style-type: none"> 標準装備のVNC (Virtual Network Computing) / RDP (Remote Desktop Protocol) を使用したロボットの修正・検証 サードパーティ画面共有ツール (WebEx等) 上での修正・検証も可能
オフライン保守	<ul style="list-style-type: none"> 実行時エラーのスクリーンショットを使った修正・検証 ローカル保存したHTMLファイル一式を使った修正・検証
イメージコレクション強化	<ul style="list-style-type: none"> ロボットが検知出来なかったイメージをテンプレートへ追加することで停止リスクの軽減が可能 必要最低限のコード修正 (条件文の追加が不要)
容易な別環境へのロボット移行	<ul style="list-style-type: none"> 移行先環境 (別OS・別解像度) で取得したイメージを既存のテンプレートと置き換えで対応が可能