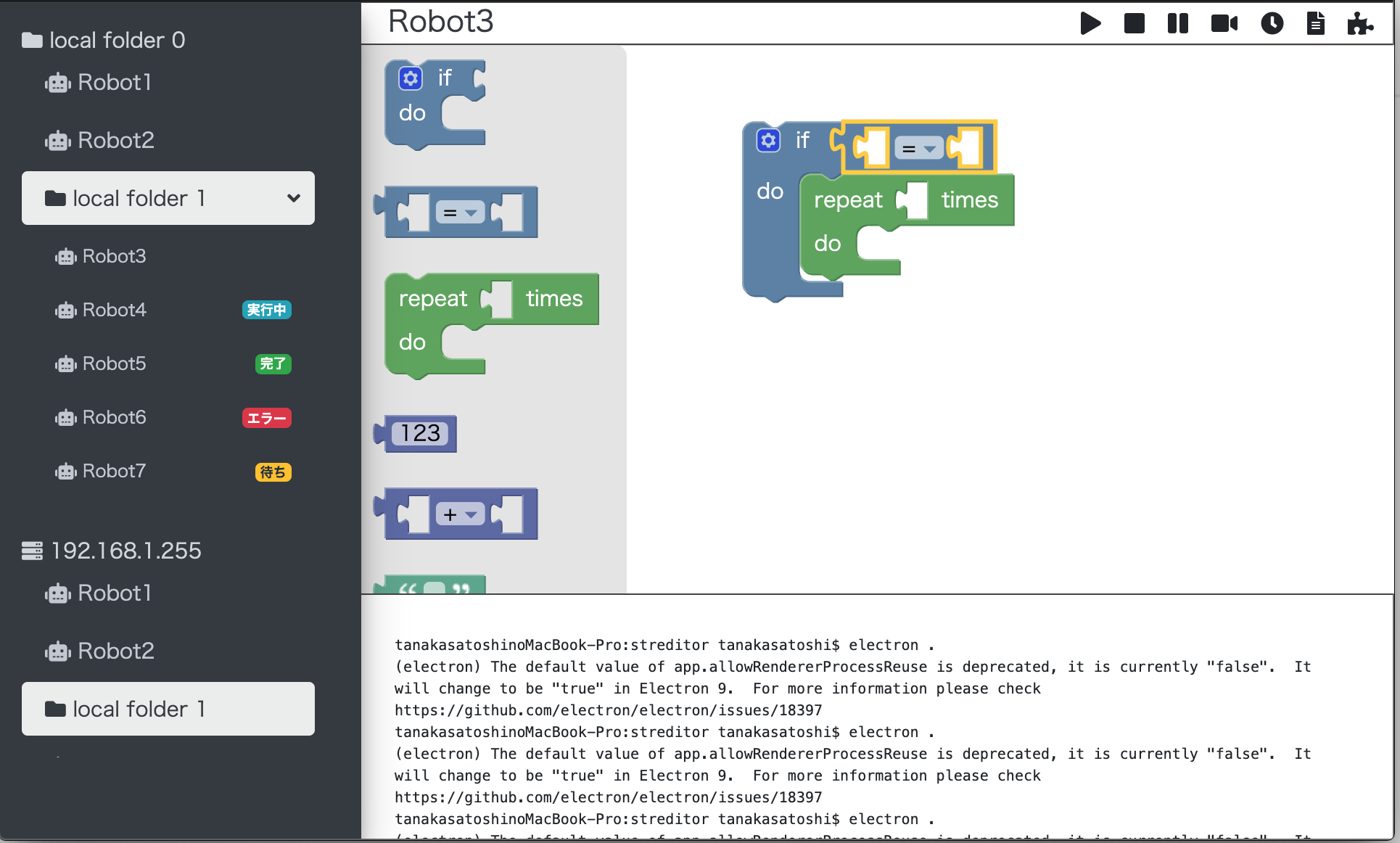
# ステラロボ設計

RPAの統合開発環境（エディター）、管理ソフトウェア（ランチャー）。

プロジェクト管理：<https://globalpocket.atlassian.net/jira/software/projects/STLROB/boards/9/roadmap>

## イメージ



## 開発計画

* 初期は現行モデルの以下の機能をラッピングして開発を行う。※このため以下は開発範囲外。
  + エンジン
  + レコーダー
* エディターがないと肝心のロボが作成できないので、エディターから開発を行う。
* 初期はフルスクラッチをするほどの長期開発は不可能なので、積極的にオープンソースライブラリ等を採用して工数の削減を行う。
  + ただし、採用するライブラリはよく吟味の上で、特に開発元の信頼性とライセンスには気をつける。

## 基本設計

当プロジェクトの直近スケジュールでは、以下の２つのアプリケーションを開発する。

1. エディター
2. ランチャー

両アプリの機能棲み分けは以下の通りとする。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **エディタ** | **ランチャ** | **実装バージョン** | **概要** |
| **ロボの処理内容の閲覧（テキストコード）** | ✓ | ✓ | **当バージョン** | 1. 処理内容を読み込む。 2. 処理内容をテキストエディタに表示する。 |
| **ロボの処理内容の閲覧（ブロック）** | ✓ | ✓ | 次期バージョン |  |
| **ロボの実行** | ✓ | ✓ | **当バージョン** | 1. 処理内容を読み込む。 2. 処理をロボに依頼する。 |
| **ロボのデバッグ実行** | ✓ |  | 未定  （当バージョン希望） |  |
| **ロボのスケジュール実行** |  | ✓ | **当バージョン** | 1. スケジュールを登録する。 2. スケジュールに従いロボの実行。 |
| **ロボの実行済みログの閲覧** | ✓（スタックトレース） | ✓ | 未定  （当バージョン希望） |  |
| **ロボの実行中状態の閲覧（テキストコード）** | ✓（簡易） | ✓ | 次期バージョン  （可能であれば当バージョン） |  |
| **ロボの実行中状態の閲覧（ブロック）** | ✓（簡易） | ✓ | 次期バージョン |  |
| **テキストエディタでのロボの作成、編集** | ✓ |  | **当バージョン** | **新規作成する。**  新規作成時テンプレートファイルを指定して処理内容を読み込む。  **既存ファイルの処理内容を読み込む。**  ファイルを指定して処理内容を読み込む。  **テキストエディタに表示する。**  処理内容の読み込みや変更を検知して処理内容をテキストエディタに表示する。  **テキストエディタで編集する。**  編集されたテキストエディタの内容を処理内容に反映する。  **保存する。**  処理内容を指定ファイルに保存する。 |
| **ブロックエディタでのロボの作成、編集** | ✓ |  | 次期バージョン |  |
| **レコーディングによるロボの作成** | ✓ |  | **当バージョン** | 1. レコーダーにレコーディングを依頼する。 2. レコーダーの処理終了を待つ。 3. レコーダーから処理内容を受け取り、現在の処理内容に上書きする。 |
| **ローカルロボのリスト表示** |  | ✓ | **当バージョン** | 1. 指定フォルダのフォルダツリー情報を再帰的に取得する。 2. フォルダツリー情報をエクスプローラーに渡す。 |
| **別マシンロボのリスト表示** |  | ✓ | 未定  （可能であれば当バージョン） |  |
| **複数のロボの状態監視** |  | ✓ | 次期バージョン |  |
| **ロボファイルの名称変更** | ✓ | ✓ | 未定  （可能であれば当バージョン） |  |
| **ロボファイルの複製** | ✓ | ✓ | 未定  （可能であれば当バージョン） |  |
| **ロボファイルの削除** |  |  | 未定  （可能であれば当バージョン） |  |

## 使用するライブラリについて

当プロジェクトでは以下のライブラリを使用する。

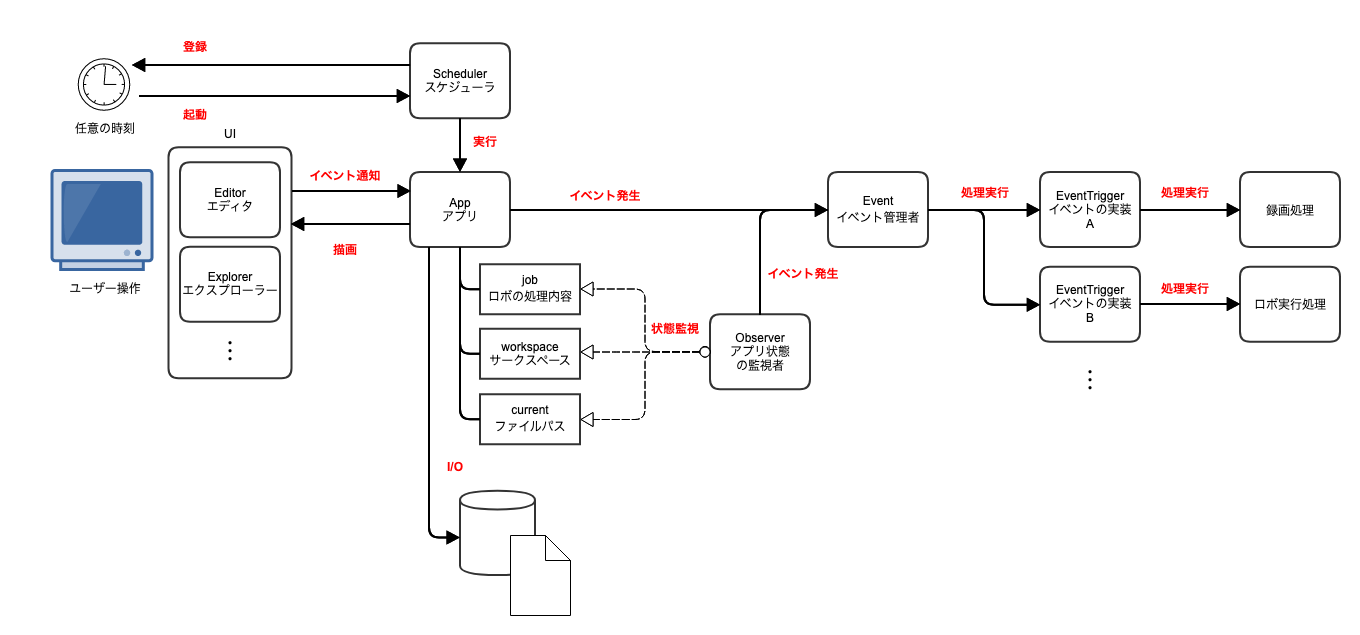
|  |  |  |
| --- | --- | --- |
|  | **概要** | **ライセンス** |
| **Node.js** | JSでアプリが開発できる統合開発環境。 | MITライセンス |
| **Electron** | VisualStudioにも採用されているウインドウ等を提供するプラットフォーム。Chromium。 | MITライセンス |
| **node-schedule** | スケジューラーライブラリ。 | MITライセンス |
| **AdminLTE** | BootStrap Project 提供のGUIライブラリ。 | MITライセンス |
| **BootStrap** | Twitter社提供のGUI デザインフレームワーク。AdminLTEの依存ライブラリ。 | MITライセンス |
| **Blockly** | Google・MIT製ブロック式エディタ。 | Apache 2.0ライセンス |
| **Monaco-Editor** | VisualStudio Codeに採用されているテキストエディタ。 | MITライセンス |

### 各ライセンスの必要事項

両ライセンスともに以下の必要事項が存在する。

1. 納品時ドキュメントとして作者の提示する免責事項ドキュメントを付属すること。
2. 「このアプリについて」と納品時ドキュメントに以下を表示。
   * 著作権の表示

## 基本構造



## 実装規約

1. HTMLにはJSを記載しない。
2. エラーハンドリングはエントリポイントにて行う。
3. プロパティ変数はgetter、setterを実装し、ログを出力する。
4. 処理とイベントは最初に引数の妥当性チェックを行う。妥当でない値だった場合は例外を発行する。
5. 処理とイベントは最初と最後にログを必ず出力する。戻り値がある場合はその内容も出力する。

## 機能開発について

### 当バージョン：０．１

#### グローバル：アプリケーション情報 App

アプリケーションの情報エンティティ。値は常に永続化される。以下の項目を持つ。

##### 変数：処理内容 App.job

ロボの処理内容。

##### 変数：ワークスペースのパス App.workspace

参照するフォルダのパス。

##### 変数：現在のファイルのパス App.current

現在参照しているファイルのパス。

##### 処理：処理内容を読み込む。 App.readJob(path)

1. 指定ファイルの処理内容を読み込む。

##### 処理：エディタを切り替える。 App.changeEditor(Editor.Mode.Text) App.changeEditor(Editor.Mode.Visual)

1. エディタモードを指定のエディタにする。
2. エディタを表示する。

##### 処理：処理をロボに依頼する。 App.runJob(function(result){}) 実装済

1. 処理内容を指定してロボを起動する。
2. 処理終了時、指定の処理を実行する。

#### グローバル：イベント Event

1. 当アプリ全体のイベントに関する実装。イベント発火のみを実装範囲とする。
2. イベント名の固定値を持つ。
3. 外部イベントを内部イベントへ変換する。
4. オブザーバーからの通知をイベントへ変換する。

#### グローバル：オブザーバー Object.observe(App, function(changes){})

1. アプリケーション情報の変更を監視する。
2. アプリケーション情報の変更を検知した場合、アプリケーションに通知する。
3. 通知は以下の通り。
   1. 処理内容を変更 **Event.App.job**
   2. ワークスペースのパスを変更 **Event.App.workspace**
   3. 現在のファイルのパスを変更 **Event.App.current**

#### グローバル：スケジューラー Scheduler

##### スケジューラーは以下を実装する。

##### 処理：スケジュールを登録する。 Scheduler.scheduleJob(date,path)

1. スケジューラーに指定の日時と任意のファイルを登録する。

##### イベント：スケジュール実行 Event.Scheduler.**onschedule**

1. 予定された処理をロボへ依頼する。

#### グローバル：エディタ Editor

エディタ（ビジュアルエディタ、テキストエディタ）は以下を実装する。

##### 処理：表示する。 **Editor.bind(source)**

1. 指定内容をエディタに表示する。

##### イベント：新規作成 **Event.Editor.new**

1. UIから発火する。
2. 新規作成時テンプレートファイルを指定して処理内容を読み込む。

##### イベント：既存ファイル読み込み **Event.Editor.read**

1. UIから発火する。
2. ファイルを指定して処理内容を読み込む。

##### イベント：編集 **Event.Editor.edit**

1. UIから発火する。
2. 編集されたエディタの内容を処理内容に反映する。

##### イベント：保存 **Event.Editor.save**

1. UIから発火する。
2. 処理内容を指定ファイルに保存する。

### グローバル：エクスプローラー Explorer

エクスプローラーは以下を実装する。

##### 処理：フォルダツリー情報を取得する。 **Explorer.**getDirTree(path)

1. 指定のフォルダパス以下のツリー情報を取得して返す。

##### 処理：表示する **Explorer.bind(dirTree)**

指定の内容をエクスプローラーに表示する。

##### イベント：ファイル選択 **Event.Explorer.selected**

1. UIから発火する。
2. 現在のファイルのパスを、選択されたファイルのパスに変更する。

## 以降今後の対応

バージョン０．１の開発に注力し、余力で優先順に以下を開発する。

### 未定（当バージョン希望）：０．２

#### ロボのデバッグ実行

#### ロボの実行済みログの閲覧

### 未定（可能であれば当バージョン）：０．３

#### 別マシンロボのリスト表示

#### ロボファイルの名称変更

#### ロボファイルの複製

#### ロボファイルの削除

### 次期バージョン（可能であれば当バージョン）：０．４

#### ロボの実行中状態の閲覧（テキストコード）

### 次期バージョン：０．５

#### ロボの処理内容の閲覧（ブロック）

#### ロボの実行中状態の閲覧（ブロック）

#### ブロックエディタでのロボの作成、編集

#### 複数のロボの状態監視