

1. 担当 PM

石黒 浩 PM

(大阪大学 大学院基礎工学研究科 システム創成専攻 教授 (特別教授))

(ATR 石黒浩特別研究室室長 (ATR フェロー))

2. 採択者氏名

チーフクリエータ：馬場 匠見 (慶應義塾大学 大学院政策・メディア研究科)

3. 委託金支払額

2,304,000 円

4. テーマ名

実世界プログラミングのための分散人力処理環境の開発

5. 関連 Web サイト

<https://github.com/babascript>

<http://babascript.org/>

6. テーマ概要

実世界における仕事の多くは人が判断し実行しているが、その際にミスは起こりがちである。自らの経験則によってミスの発生を回避することは可能だが、その熟練した経験則を得ることは容易ではない。一方で、これらの仕事の多くはマニュアル化されたものであり、プログラムとして記述可能な点が多い。ならば、仕事の実行主体である人間をプログラムモジュールとして利用可能にすれば、仕事をプログラムとして実行できるようになると考えられる。このよう

な環境下では、プログラム通りに仕事を厳密に実行していくことになり、人は人でしか実行できないような仕事に集中し、ミスを抑え、効率的に仕事を実行していくことが可能となる。

そこで本プロジェクトでは、上記のように人がプログラムからの指令に従うことで、人が経験則などに左右されず業務の遂行や目的の達成が可能となるような仕組みを実現する。そのために、プログラム上で人をモジュールとして扱えるようなプログラミング環境を開発する。具体的には、プログラムから人に対して命令を送れるライブラリと、プログラムからの命令を受け取り、人が処理した結果をプログラムに返すようなアプリケーションの実装をする。また、分散処理の仕組みを取り入れ、多数の人を対象とした命令配信を可能にする。

本プロジェクトの仕組みにより、人間からの返り値と実世界情報などを組み合わせ、より柔軟な人・実世界のプログラミングが可能になるため、様々な応用が期待される。

7. 採択理由

コンピュータに指令を与え思い通りに動かすというのが、これまでのプログラミング言語の常識であるが、本提案では、指令を与える対象をコンピュータではなく、人間にしている。すなわち、”人” または”人々” に対する様々なコマンドを持つプログラミング環境において、人間の行動をプログラムするという野心的なアイデアを提案している。

この野心的なアイデアを実現するためには、人間をどのように制御するか、どのような環境を対象とするか等、慎重に検討しなければならない項目は幾つもあり、それらの慎重な検討が成功の鍵であると同時に、容易な検討ではないとも想像される。しかしながら、そうした困難を認識した上でも、本提案の独創的な発想は魅力的であり、チャレンジする価値は十分にあると認められる。また一定レベルの完成度を得られたなら、様々な形で本手法の可能性が広がることも期待される。

8. 開発目標

本プロジェクトは、プログラム上において人とコンピュータの双方を対等に扱い、処理内容に応じて適切なノードに処理を行わせられるような仕組みの実現と、その仕組みを利用し、人の仕事や役割をプログラム化することを目的とした。

この目的を達成するために、人への行動指示をプログラム上で記述可能にす

る Babascript プログラミング環境を構築し、本環境を利用した人の仕事や役割のプログラム化に取り組むことで、Babascript の応用可能性について考察するものとした。

9. 進捗概要

Babascript、Babascript Client、Babascript Manager、Babascript Plugins を開発した。

Babascript は、人への行動指示と指示に対する返り値を得ることのできるプログラミングライブラリであり、Node.js で動作するように実装した。Babascript は、人への行動指示命令を取得し、タスクへと変換する人への命令構文と、分散処理基盤との通信を行う機能、指示に対する人の実行結果を返り値として受け取る機能、プラグイン機能などを有し、通常のプログラミング記法とほぼ同じ記法で人をプログラムに組み込むことができる。これによって、人を実世界とのインタフェースとして利用可能となる。例えば、図 1 のようなプログラムによって人をプログラム上に宣言し、行動指示命令を送ることができる。

```
1  Baba = require "babascript"
2
3  baba = new Baba "baba"
4  baba.成果報告書を書く {}, (result) ->
5  console.log result
6  #result = {
7    #value: "true"
8    #task:
9      #key: "成果報告書を書く"
10     #name: "baba"
11     #type: "eval"
12     #cid: "1403501688\_0.2959382198750973"
13     #at: 1403501771342
14     #options: {}
15     #getWorker: ->
16     #return new Baba "baba"
17  #}
```

```
1  member = new Baba "2013mitoh"
2  member.成果報告書を書く {format: 'string', broadcast: 17}, (results) ->
3  console.log results.length # 17
4  for result in results
5    # ...
```

```
1  Baba = require "babascript"
2  UserManager = require "babascript-plugin-usermanager"
3
4  baba = new Baba "baba"
5  baba.set "manager", new UserManager()
6  # ...
```

図 1 Babascript プログラム例

Babascript Client は、Babascript からの命令を受け取り、命令を人に見える形に変換・提示し、返り値を入力させ、プログラムに返すための一連の機能を実装したライブラリとアプリケーションであり、Node.js、Web アプリ、Cordova アプリ、iOS アプリとして実装した。Babascript Client は、図 2 のようなプログラムで Babascript からの命令を受け取ることができる。

```

1 Client = require "babascript-client"
2 client = new CClient "baba"
3 client.on "get_task", (task) ->
4   #... 命令を取得、処理を記述する
5 client.on "cancel_task", ->
6   #... 命令がキャンセルされた時の処理を記述

```

図 2 Babascript Client プログラム例

インタフェースは図 3 のように実装した。命令の受信や返り値の送信をする機能とは分離した実装であるため、既存の UI だけでなく、他メディアをインタフェースとして利用する、といったことも可能である。



図 3 Babascript Client インタフェース

Babascript と Babascript Client を組み合わせることで、プログラム上において人を表現し、その人に対する行動指示命令と返り値の取得が実現する。

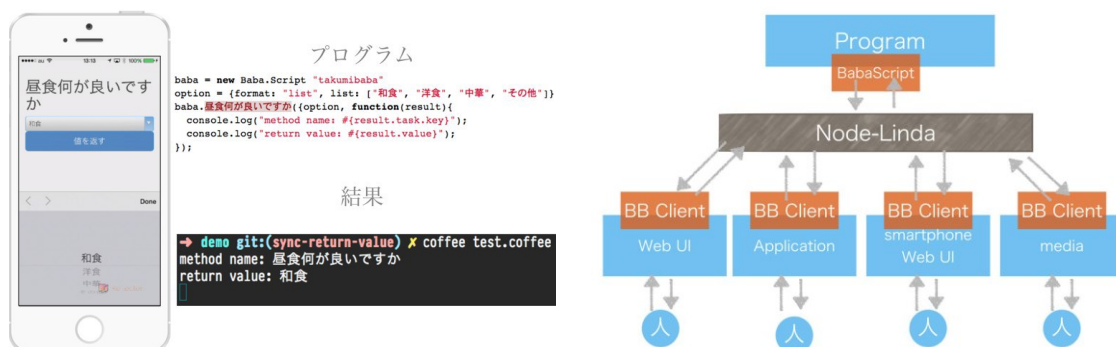


図 4 Babascript 全体像とシステム図

Babascript Manager は、Babascript で利用するための人カリソースを管理可能にする Web サービスである。ユーザ・グループ情報の管理や、ユーザ情報のリアルタイム同期、タスクログの保存などの機能を有する。

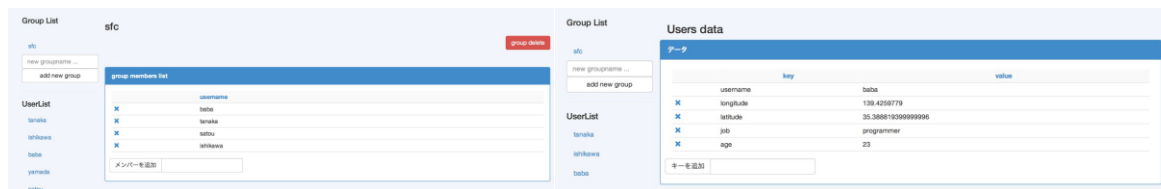


図 5 Babascript Manager Web インタフェース

Babascript は、プラグインによってその機能を拡張していくことができる。そこでプラグインの例として、UserManager プラグインと Logger プラグインの 2 つを実装した。UserManager プラグインは、前述の Babascript Manager と Babascript 及び Babascript Client を接続するためのプラグインだ。Logger プラグインは、人への行動指示命令や取得した返り値をログとして保存するためのプラグインである。プラグインは、指定されたメソッドを実装したオブジェクトを読み込むだけで簡単に Babascript 及び Babascript Client に組み込むことができる。

```
10 baba.data.users.get("baba").on "change_data", (data) ->
11   if data.level > 10
12     baba.data.users.get("baba").set "job", "master", {sync: true}
13     baba.研究してください {}, (result) ->
14       console.log baba.data.users.get(id).get 'job'
15       console.log result
```

図 6 Babascript プラグイン

また、Babascript 環境を利用して人の仕事や役割をプログラム化し、実行するといったことに取り組んだ。仕事や役割をプログラム化することによって、コンピュータにできることはコンピュータに実行させ、人は人にしかできないようなことを実行させることが可能となり、人への負担軽減やヒューマンエラー防止に繋げることができる。また、プログラム化対象を検討し、「論文を読む」「料理する」「ミーティング自動化」「スケジュール調整」といった、プログラムに近い仕事をプログラム化し、実行した。結果として、逐次的指示や作業の分割提示などが良いという評価を得るなどした。また、人の処理の遅れがプログラム実行や処理継続に大きな影響を与えることや、命令の抽象度の適切な設計が必要であること、プログラムから命令されることに不安を覚える人がいる、といった知見を得る事ができた。

10. プロジェクト評価

人間をプログラムするという非常にユニークなアイデアの基にプロジェクトに取り組んだ。そして、必要なコマンドを着実に実装すると共に、自分の周りの人間を参考に、実際に人間制御用のプログラムを幾つも作ったことは高く評価できる。

しかしながら残念なのは、本来非常に高いポテンシャルを持つシステムにもかかわらず、採択者の人生経験の浅さから、真に役に立つ人間制御用のプログラムは実現できなかったようにも思える。プロジェクト期間終了後も是非とも開発を続けて、本当にプログラム可能なケースを見つけてほしい。

11. 今後の課題

今後の課題として、多言語対応、プラグイン機構の改善、サンプルプログラムの拡充などが挙げられる。また本システムをオープンソースのプロジェクトとして公開し、多くの人に使ってもらえるような体制を整える必要がある。