Babascript:

Babascript:

匿名で査読を行うため著者名なし *

abstract

1 はじめに

あらゆるモノ・コトがプログラムで記述されるような未来が近づいているプログラムで処理できることは増え続けている実世界においても、それは言えること実世界志向インタフェースなどの研究は多くの成果を出している Hue などの商品によって、実際に誰でもプログラムできるようになってきているこの流れは変わらず、全世界がプログラム可能になっていくと考えられる。

一方で、コンピュータのみでは処理できないようなこともある人の確認が必要な状況 人の意志の反映が求められる状況人でなくてはいけないような状況こういった問い合わせはプログラムに組み込みづらい結果として、プログラム化・自動実行などが難しくなっている。

コンピュータの方が得意なこともあれば、人のほうが得意なこともある何かしらの処理を実現する上で、コンピュータと人は相補的に動作していくべきである今までは出来なかったようなことの実現

上記の状況を踏まえ、本論文ではプログラムから人へ、処理命令を送れる人力処理組み込み環境 Babascript を提案する。 Babascript 環境では、プログラムというフォーマット上においてコンピュータと人、双方への命令を同様のインタフェースで記述可能になる特殊なプログラミング言語を導入せず、既存の言語上において実現可能

コンピュータはプログラムという手順書を実行する人も、常日頃から手順書に基づいて行動している ex: マニュアル、レシピ手順書のフォーマットが異なるだけで、処理内容が記述された手順書を参照していることには変わらないつまり、手順書のフォーマットは同一にすることも可能である。

Copyright is held by the author(s).

* 匿名で査読を行うため所属名なし

2 Babascript

Babascript 環境は、以下の 2 点を基本構成としたプログラミング環境だ。

- Babascript
- Babascript Client

これに加え、基本構成の機能を補完するプラグイン機構によって拡張していくことが可能だ。

Babascript 環境では、プログラムに人を組み込めるようにするために、関数実行によって人に命令を送れる仕組みを持つ。従来のプログラミング環境においては、プログラムからコンピュータに対して処理を命令し、処理結果を値として得ている。Babascript 環境では、上記のフローと同じように、プログラムから人に対して処理を命令し、処理結果を値として得られる。関数というインタフェースを通して、人とコンピュータに対する命令をプログラム上においてほぼ同一のものとすることによって、プログラム上において簡単に人を扱えるようにする。

以下のような流れで利用可能である。

1. 人への命令構文を実行する
2. 命令構文を元にタスクを生成する
3. 分散処理基盤を通して適切なクライアントへタスクを配信する
4.
5.
6. 7.

2.1 Script

Babascript は、人オブジェクトをプログラム上で扱えるようにした DSL だ。人オブジェクトにおいて定義されていない全てのメソッドが人への命令として解釈され、タスクが生成される。また、命令に対して人からの返り値を得ると、実行メソッドの引数で指定したコールバック関数を実行する。

```
baba = new Baba.Script "baba"
baba.ほげふが (task) ->
  console.log 'hoge'
```

人オブジェクトは宣言時に ID を指定することによって、どのクライアントに対して処理を配信するかを決定する。指定された ID を監視するクライアントにのみ、タスク情報が配信される。

2.1.1 オプション情報の付加

2.2 Client

Babascript Client は、Babascript からの命令受信と処理結果の送信機能を実現するクライアントライブラリだ。

2.3 Plugin

Babascript Plugin は、Babascript 及び BabascriptClient に組み込むことのできるプラグイン機構だ。プラグインとして読み込んだ各種プラグインは、Babascript 及び BabascriptClient が発行するイベントとデータを受け取ることができる。現在対応しているイベントは、以下の通りだ。

- init(プラグイン組み込み時)
- connect(ネットワーク接続時)
- send(タスクの送信時)
- receive(タスクの受信時)

具体的には、以下のようなプラグインの実装が挙げられる。

- ログコレクター
- データ同期
- ユーザ管理

2.4 通信と分散処理

script と client のタスク送受信と分散配信のために、node-linda[2] を利用した。

2.5 実装

2.6 サンプルプログラム

---

3 実装

---

4 応用例

以下のような応用が考えられる。

- 人の行動をプログラミングする
- 実世界をテストする
- あれ

---

5 議論

5.1 タスク実行の遅延と実行保障性

Babascript によってタスク実行を依頼しても、人がすぐにタスクを実行し値を返すことを完全に保証することはできない。タスク受信端末を見ていない、受信しても実行できないといった状況の場合、すぐに値を返すことはできない。こういった際、Babascript による処理がボトルネックとなる可能性がある。

また、労働関係にあるなど、タスク実行に強制力がある場合は、タスク実行が確実に行われると考えられるが、強制力がない場合はそもそもタスク受信を無視するといったことも考えられる。タスク実行に強制力がない場合は、金銭などのインセンティブを与えるといった手段によって、実行保障性を確保するといったことが考えられる。

5.2 命令内容の粒度

Babascript では、タスクの文面の記述には制限がないため、自由となっている。この文面は、適切な抽象度の文面に設計しなくてはならない。抽象度が高すぎる命令は、あいまいな表記となり、タスク実行者にとって理解しづらい文面となり得る。その結果、想定外の処理が実行され、意図しない結果を招く恐れがある。抽象度が低すぎる命令は、全体の処理内容にもよるが、プログラム自体が冗長となり得る。プログラムとタスク実行者の間のやりとりが増え、通信や待機時間などがボトルネックとなる可能性がある。また、タスク実行者にとっても、やりとりが増えることで負担増になると考えられる。

5.3 同時の複数命令

複数のプログラムから同時に一人のタスク実行者へとタスクが配信される可能性がある。この際、異なるコンテキストにある命令が交互に配信され、タスク実行に大きな障害をもたらす可能性がある。例えば、料理プログラムと掃除プログラムが同時に実行された場合、鍋で煮ている途中で「洗剤を投入しろ」などといった命令が配信されることが考えられる。

この問題は、全ての Babascript プログラム中において、一人のタスク実行者は一つのプログラムからのみ、連続してタスクを受信できるような仕組みを用意することによって、解決可能であると考えられる。また、応用アプリケーションでの実装になるが、コンテキストを明示し、どの処理系におけるタスクなのかをタスク実行者に示すといった手段によっても解決可能である。

---

6 関連研究

計算機では処理が難しいようなタスクを解決するために、人を計算資源として利用する手法はヒュー

マンコンピュテーション [8] と呼ばれ、様々な研究が行われている。インターネットを介して不特定多数の群衆にタスクを実行させるクラウドソーシングと組み合わせた研究事例も多く存在する。クラウドソーシングのプラットフォームとしては、Amazon Mechanical Turk[1] が存在する。 Barowy らは、CrowdProgramming という概念を提唱し、プログラミング言語内においてクラウドソーシングによる計算とコンピュータによる計算の統合を実現した [3]。Franklin らは、機械だけでは答えられないような DB へのクエリに対する応答を、クラウドソーシングを用いることで返答可能にする CrowdDB を提案している [6]。Morishima らは、人をデータソースとしてプログラムの中で利用する手法を提案している [7]。jabberwocky crowdforge これらの研究では、人を計算資源やデータソースとしてシステムに組み込むことを狙っている。本研究では、計算資源やデータソースに限らず、実世界への干渉等も対象としており、本研究はより汎用的な枠組みとなっている。また、いずれもクラウドソーシングの利用を前提としているが、本研究はクラウドソーシングを対象としたものではない。

ユビキタスコンピューティングの研究分野においては、Human as Sensor という概念も提唱されている。PRISM は、スマートフォンを利用したセンシングプラットフォームだ [5]。Liu らは、ソーシャルメディア上の人をセンサーとして扱った Q&A サービス MoboQ を提案し、その検証を行った。Human as Sensor に類する研究では、人をセンサーとして扱うことを対象としているが、本研究ではセンサーのみを対象としていない。

Cheng らは、人をモーションプラットフォームにおけるモーターやメカニカル機構の代替として利用した Haptic Turk を提案している [4]。 Haptic Turk はゲームでの利用に特化したものだ。本研究は、使用用途を限らない汎用的な仕組みとなっている。

加藤らは、人とロボット間でのタスク共有システム Sharedo を提案した [9]。人とロボットのタスク実行における協調は、本研究の主眼である「何かの処理を実現するとき、人とコンピュータは相補的に動作できるべき」という考えと大きく類似している。

7 おわりに

本論文では、人への命令構文をプログラムに付与可能なプログラミング環境 Babascript を提案した。Babascript 環境においては、プログラム上において人は、コンピュータと同じ処理ノードとして存在し、関数実行によって処理内容を受け取り、実行・値を返す存在になれる。これによって、プログラマブルになりつつある世界において人自身もその一部になることが可能となる。

また、Babascript 環境によって実現する応用例を示すことによって、人をプログラマブルにした際のメリットを示した。

今後は、議論で述べた Babascript の問題点などを改善していく。

参考文献

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/.

[2] Linda implementation on Socket.IO. https://github.com/node-linda/linda.

[3] D. W. Barowy, C. Curtsinger, E. D. Berger, and A. McGregor. AutoMan: A Platform for Integrating Human-based and Digital Computation. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications*, OOPSLA '12, pp. 639–654, New York, NY, USA, 2012. ACM.

[4] L.-P. Cheng, P. Lühne, P. Lopes, C. Sterz, and P. Baudisch. Haptic Turk: A Motion Platform Based on People. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 3463–3472, New York, NY, USA, 2014. ACM.

[5] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. PRISM: Platform for Remote Sensing Using Smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pp. 63–76, New York, NY, USA, 2010. ACM.

[6] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering Queries with Crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pp. 61–72, New York, NY, USA, 2011. ACM.

[7] A. Morishima, N. Shinagawa, T. Mitsuishi, H. Aoki, and S. Fukusumi. CyLog/Crowd4U: A Declarative Platform for Complex Data-centric Crowdsourcing. *Proc. VLDB Endow.*, 5(12):1918–1921, Aug. 2012.

[8] L. Von Ahn. *Human Computation*. PhD thesis, Pittsburgh, PA, USA, 2005. AAI3205378.

[9] 淳 加藤, 大介 坂本, 健夫 五十嵐. Sharedo: To-do リストによる人-ロボット間のタスク共有. WISS '11: 第 19 回インタラクティブシステムとソフトウェアに関するワークショップ, pp. 102–107, 2011.