

# 情報処理学会論文誌ジャーナル論文の準備方法 (2012年2月4日版)

情報 太郎<sup>1,a)</sup> 処理 花子<sup>1</sup> 学会 次郎<sup>1,†1,b)</sup>

**概要:** 本稿は、情報処理学会論文誌ジャーナルに投稿する原稿を執筆する際、および論文採択後に最終原稿を準備する際の注意点等をまとめたものである。大きく分けると、論文投稿の流れと、 $\text{\LaTeX}$  と専用のスタイルファイルを用いた場合の論文フォーマットに関する指針、および論文の内容に関してすべきこと、すべきでないことをまとめたべからずチェックリストからなる。本稿自体も  $\text{\LaTeX}$  と専用のスタイルファイルを用いて執筆されているため、論文執筆の際に参考になれば幸いである。

## How to Prepare Your Paper for IPSJ Journal (version 2012/2/4)

JOHO TARO<sup>1,a)</sup> SHORI HANAKO<sup>1</sup> GAKKAI JIRO<sup>1,†1,b)</sup>

**Abstract:** This document is a guide to prepare a draft for submitting to IPSJ Journal, and the final camera-ready manuscript of a paper to appear in IPSJ Journal, using  $\text{\LaTeX}$  and special style files. Since this document itself is produced with the style files, it will help you to refer its source file which is distributed with the style files.

### 1. はじめに

実世界環境と IT システムの融合が進んでいる。実世界の状態にもとづいて Web が変化したり、Web の情報にもとづいて実世界が変化することが起きている。プログラムから実世界環境へアクセスすることは増えていっており、Arduino などによってそれは容易になってきている。ちょっとした労力で実世界をセンシングしたり、アクチュエータで実世界に干渉できるようになっている。この流れの中で、人力をプログラムから用いるという試みが行われている。人は汎用的なセンサ・アクチュエータであり、コンピュータでは難しいようなことでも瞬時にこなしてしまう事例がある。こういった人力をプログラムに組み込むことで、より強力なコンピューティングが可能となる。しか

し、現状のこれらの取り組みのほとんどはクラウドソーシングなどの世界中に散らばる不特定多数を対象としているものだ。特定可能な身近な人や実世界環境を対象としているものはなく、コンシューマや一般開発者にとって人力を使うことは難しい。Arduino などのツールがプログラムから実世界を扱うことを一般開発者でも出来るようにしたように、人力を利用するための環境も一般開発者でも使えるようなレベルに落としこむ必要がある。そこで、本論文では、身近な人間を組み込んでプログラミングするための手法を提案する。また、その応用例を示す。

### 2. 分散人力処理環境

人をプログラムに組み込むための環境について論じる。人をプログラムに組み込むためには、以下の要素が必要だ

- 人への命令構文
- 処理した結果をプログラムに返す

つまり、人への命令構文を通常の関数実行と同じ形式にし、同様に値が返ってくるようにする必要がある。そこで、以

<sup>1</sup> 情報処理学会  
IPSJ, Chiyoda, Tokyo 101-0062, Japan

<sup>†1</sup> 現在、情報処理大学  
Presently with Johoshori University

<sup>a)</sup> joho.taro@ipsj.or.jp

<sup>b)</sup> gakkai.jiro@ipsj.or.jp

下の2つを開発した。

- BabaScript
- BabaScript Client

また、適切なクライアントへ命令を分配するために、Linda-Base という分散処理プラットフォームを導入した。

## 2.1 BabaScript

BabaScript は、プログラム言語に人への命令構文を追加可能なライブラリだ。Ruby 上で動作するよう実装した。BabaScript 及びそれを継承したクラスで宣言されていないメソッド全てが人への命令メソッドとして扱われる。命令メソッドとして扱われた場合、メソッド名と引数が人に通知される。引数内で指定した内容に応じて、BabaScript Client の挙動を変化させることができる。例えば、第一引数の `:format` に `"int"` を指定すれば、BabaScript Client では `int` 型の入力フォームが表示される。

```
1 num = baba.研究室にいる人の数は({:format => "int"}, "デルタ")
2
3 if num > 3
4   baba.夕飯は鍋にしましょう()
5 else
6   baba.夕飯は学食に行きましょう()
```

図 1 キャプション

## 2.2 BabaScript Client

BabaScript Client は、プログラムと人とのインタフェースとなるクライアントアプリケーションだ。スマートフォンアプリケーションとして実装し、Android 上で動作する。以下の役割を持つ

- プログラムから人への命令を受け取る
- 人に命令内容を伝え、結果を入力させる
- 入力された値をプログラムに返す

ユーザは、スマートフォン上に表示されるプログラムからの命令に返答する。返り値はプログラムから指定されたものを使う。例えば、`boolean` を指定された場合、処理結果の入力フォームは `true` or `false` の二択となる。

## 2.3 分散処理の仕組み

人や人たちへの命令を適切に配信するためのプラットフォームとして、Linda-Base という分散処理環境を導入した。Linda-Base は Linda という分散処理のための仕組みを Web 経由で利用可能にした web サービスだ。命令は Tuple というものでやりとりされ、Tuple の最初の項目に書かれている内容に応じて命令を振り分ける。例えば、`"takumibaba"` という内容であれば、`"takumibaba"` を監視しているクライアント全てに命令が振り分けられる。

## 3. 応用例

本提案の人力処理環境を応用すると、以下のようなことが可能となる。

### 3.1 人の役割/仕事のプログラム化

人は様々な役割や仕事を担っている。その中には、決まりきったタスクをこなすだけであったり、マニュアルとして言語化されているものがある。人の行動が一定で言語化可能であるならば、本提案の環境を利用することで、それらの役割/仕事をプログラムとして記述可能になる。プログラムとして実行することで、人の意思を必要としないような条件判断などをプログラムに任せ、人は人にしかできないような仕事に専念することができる。また、ただプログラムからの命令を実行していただけなので、素人でも業務遂行/役割達成が可能となる。以下のようなメリットが考えられる。

- 引き継ぎ作業が必要ない
- 効率的に人が運用可能
- 作業状態の保存が可能
- 一方的に指示してもらえる

ただ命令に従うだけなので、人員交代などによる引き継ぎがあっても問題ない。人の代替が可能。人の作業状態をプログラムで管理することができるので、A さんがやっていたことを B さんにやらせる、といったことが簡単にできるようになる自分のすべきことは基本的にプログラムが教えてくれるので、それに従うだけ。

### 3.2 エラー処理の人への委譲

### 3.3 移動型のセンサー・アクチュエータ

## 4. 関連研究

HumanComputation クラウドソーシング分野 Human as Sensor など Jabberwocky AutoMan CrowdDB プログラムに人を組み込むという類似例としては、CrowdDB が挙げられる。CrowdDB はコンピュータにはわからないようなクエリを人間が処理させるものだ。CrowdDB はデータベースとの連携に主眼を置いているが、本提案は人をプログラムに組み込むことで実世界に干渉したりすることを主眼においている。

## 5. おわりに

本論文では、人をプログラムに組み込む仕組みについて提案した。また、その環境上で動くアプリケーションの利用例を示した。