

表計算インターフェースを用いたクラウドソーシング記述手法の提案

鈴木 陸也†

阪口 哲男‡

森嶋 厚行‡

† 筑波大学 情報学群 情報メディア創成学類

‡ 筑波大学 知的コミュニティ基盤研究センター

1 はじめに

近年、計算機ネットワークの発達により、不特定多数の人に仕事を委託するクラウドソーシングと呼ばれる手法が注目されている。その一つの形態として、比較的短時間で作業ができるタスクを委託するマイクロタスク型クラウドソーシングがある。マイクロタスク型に関しては Amazon Mechanical Turk や Yahoo!クラウドソーシングなどのクラウドソーシングプラットフォーム(以下 CS プラットフォーム)が存在し、簡単な作業でマイクロタスク(以下タスク)を登録可能である。これらの CS プラットフォームでは、一般に、タスクを登録し、作業の終了後にデータをダウンロードする事で、結果を入手することができる。

しかし、これらの CS プラットフォームでは、複数種類のタスクを連携させた結果が欲しい場合に作業が煩雑となる。例えば、人が写った複数の写真がある場合に、写真ごとに(1)写真に写った人間の数を数えるタスク A を委託し、次に(2)その結果の確認を行うタスク B を委託する事を考える。この場合、タスク A を登録し、結果をダウンロードし、その結果を基にタスク B を作り、それを登録し、結果をダウンロードする。ということになる。さらに、同じタスク A を複数人で作業して異なる結果が出た場合のみタスク B を行い、最終的に写真に写った総人数を集計したいといった場合には、結果の取捨選択や集計などに、専用のプログラムを書いたり、表計算ソフトウェアに取り込んだりといった作業が必要となる。

本稿ではこのような煩雑さの解消を目的として、広く利用されている表計算ソフトウェア上でこれらのワンストップサービスを提供するシステム“CrowdSheet”を提案する。CrowdSheet は、表計算ソフトウェアの関数を利用してタスクを発行するというアプローチにより、使い慣れた表計算ソフトウェアの作法を維持しつつ、プログラミングスキルのない利用者にもより容易なクラウドソーシングの利用を可能にする。

本稿では、CrowdSheet を説明し、その表現力につい

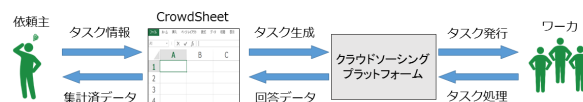


図 1: CrowdSheet のアーキテクチャ

	A 関数の挿入	B
1	=hINPUT("あなたの年齢は?")	=hINPUT("あなたの年齢は?")
2		=COUNTIF(A1:B1, ">20")

図 2: 利用例

て議論する。

関連研究. 表計算ソフトウェアとクラウドソーシングを組み合わせたシステムとしては AskSheet [1] がある。これは、比較表的作成のためにマイクロタスクを生成・発行するシステムである。また、表計算ソフトウェアではないが、エディタの裏側にクラウドソーシングを組み合わせたシステムとしては Soylent [3] がある。これは、文章作成をサポートするためのマイクロタスクを裏側で生成・発行する。CrowdSheet は目的を限定しておらず、複雑なクラウドソーシングを明示的に記述する機能をワンストップで提供する汎用システムであるという点で、これらとは全く異なるものである。

2 提案システム“CrowdSheet”

CrowdSheet のアーキテクチャを図 1 に示す。まず、利用者が表計算ソフトウェアにクラウドソーシング記述を行う。これには後で説明するタスク発行関数を用いる。次に、記述された内容に応じてシステムはタスクを生成し、CS プラットフォームに投入する。複数種類のタスクを連携させる指示がある場合には、記述にしたがって、指示通りの順にタスクが実行される。結果は最終的に表計算ソフトウェア上に現れる。

タスク発行関数. タスク発行関数とは、引数に応じてタスクを発行し、ワーカの回答がその関数の値となるものである。タスク発行関数は複数種類存在する。この種類のことをタスク型と呼ぶ。一例として、次がある。
hINPUT(String txt): txt を質問文とし、テキストフォームに入力させるタスクを発行し、その結果を返す。
hCHOICE(String txt, Range choice): txt を質問文とし、choice で指定された範囲のセルの値を回答の選択肢とするタスクを発行し、その結果を返す。

図 2 に利用例を示す。ここでは、セル A1, B1 のタス

ク発行関数から2つのタスクが生成・発行され、2つの結果の集計結果(20以上の数)がセルB2に出力される。

タスク発行関数は、他のタスク発行関数が入力されたセルを引数に指定することで、他のタスクの結果を利用したタスクを発行することが出来る。その際、表計算ソフトウェアの関数と同様にセル間の依存関係に基づいて処理順序を決定する。前述した、タスクAの回答が正しいかを確認するタスクBを作成するとき、タスクAの処理が行われてから、その結果を利用して確認のためのタスクBを発行する処理を行う。

実装. 表計算ソフトウェアとCSプラットフォームとしては、それぞれMicrosoft ExcelとCrowd4Uを利用し、タスクの生成ロジックはExcelマクロで実装した。

3 CrowdSheetの表現力の分析

CrowdSheetの表現力を議論するために、クラウドソーシングのためのプログラミング言語CyLogを用いる。CyLogはDatalogに似た表記法でクラウドソーシングの処理を記述可能なプログラミング言語である。実装システムでは、タスク発行のためにCyLogプログラムを生成する。したがって、CrowdSheetが生成するCyLogプログラムの制約を議論することが、CrowdSheetの表現力を議論することになる。

タスク発行関数から生成されるCyLogコード. セル xy に t 型のタスク発行関数が記述されたとき、タスク発行のために生成されるCyLogコードは次である。

Pattern 1

```
1: Seed(xy, type); Data1(xy, d1); Data2(xy, d2); ...;
2: Result(xy, t, result)/open <- Seed(xy, type : t);
3: !Taskt(xy, d1, d2, ...)
   <- ?Result(xy, result, t), Data1(xy, d1), Data2(xy, d2); ...;
```

CyLog概要. CyLogプログラムはDatalogと同様にファクトとルールによって記述される。ファクトはタプルの定義である。Pattern 1の1行目には複数のファクトが記述されている。

ルールは、右辺の条件を満たすデータが存在するとき左辺のデータが存在することを示す。2, 3行目はそれぞれルールにあたる。CyLog処理系では、ルールはボトムアップに評価される。すなわち、右辺の条件が成立したとき、左辺のタプルをリレーションに挿入する。

2行目の左辺にある/openは、未束縛変数を人間に問い合わせる指定であり、ここでは右辺で指定されていない属性resultの値をワークに問い合わせることを表す。

3行目の!で始まる述語(!Task_t)は、タスクに直接対応する特別なリレーションを表す。また、?で始まる述語(?Result)は、未束縛変数を含むタプルにマッチする。

Pattern 1の詳細は次の通りである。

1行目: Seed リレーションは、タスクを発行する基となるリレーションである。発行するタスク毎に1つのタプルを挿入する。この行では、セル xy に記述されたタスク型 t のタスク発行関数として記述されたタスクを発行するためのタプルを挿入している。各 $Data_i$ には、タスク型に応じたデータ(引数で指定された質問文や選択肢)を用意する。これらのデータは xy で結びつけられる。2章で説明したような、他のタスクの結果を用いてタスクを生成する場合には、Seedタプルの生成はファクトではなくルールとなり、その条件として他のタスクの結果の有無を指定する(Pattern 1では省略)。

2行目: Seedタプルが表現するタスクの結果を受け取るためのResult リレーションのタプルを生成する。

3行目: Resultの未束縛変数をもつタプル毎にタスクの生成と発行を行う。タスク型 t に応じて必要な $Data_i$ の情報をタスク生成に利用する。

CrowdSheetの表現力. 以上のコードと表計算ソフトウェアの一般的な作法の分析から、CrowdSheetで記述可能な(一般に複数の)タスク群は次の性質を持つと推測される。

- 性質1 各タスクは1つの値を返す。
- 性質2 発行するタスクの数の最大値が決まっている。
- 性質3 発行するタスクの形式はタスク発行関数が定義されているもののみに限定される。

次の定理は、上の表現力の解析が正しい事を示す。

定理. 上記性質1, 2, 3の性質を満たす任意のCyLogプログラム P を、Pattern 1にマッピングするアルゴリズムが存在する。□

4 おわりに

本稿は表計算インターフェースを用いて複雑なクラウドソーシングを容易に記述可能なシステムCrowdSheetを提案し、その表現力の検証を行った。今後の課題としては、表計算ソフトウェアの作法と互換性を保ちながら表現力を向上する事が可能か否かの検討がある。

謝辞. 本研究の一部は科研費(#25240012)による。

参考文献

- [1] A. J. Quinn, B. B. Bederson. AskSheet: efficient human computation for decision making with spreadsheets. In Proc. CSCW '14, 1456-1466, 2014.
- [2] A. Morishima, N. Shinagawa, T. Mitsuishi, H. Aoki, S. Fukusumi. CyLog/Crowd4U: A Declarative Platform for Complex Data-centric Crowdsourcing, PVLDB 5(12): 1918-1921 (2012)
- [3] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, K. Panovich. Soy-lent: a word processor with a crowd inside. In Proc. UIST '10, 313-322, 2010.