

# Web アプリとサービスの自動連携方式の実装

小池賢一<sup>†1</sup> 大松史生<sup>†1</sup> 谷屋直隆<sup>†2</sup> 高倉晴久<sup>†2</sup>

これまでの研究で、自由な形式のスプレッドシートを使用した業務アプリケーションの開発方法を提案した。しかし、この方法ではアプリケーションとサービスの連携は個別に開発する必要があった。本研究では、スプレッドシートに付加情報を与えることにより Web アプリとサービスの連携を自動化する方式を提案し実装例を示す。

## Implementation of Automatic communication between Web Application And Web Service

KENICHI KOIKE<sup>†1</sup> FUMIO OMATSU<sup>†1</sup>  
NAOTAKA TANIYA<sup>†2</sup> HARUHISA TAKAKURA<sup>†2</sup>

In our previous study, we proposed application development method using the free format spreadsheet. But the interaction between web application and web service should be made by hands. In this paper, we propose a method of automatic communication between web application and web service using spreadsheet with additional information. And we describe an example implementation of this method.

### 1. はじめに

業務アプリケーションを開発する手法としてスプレッドシート上にビジネスロジックを記述しておき、その内容をビジネスルールエンジンが解釈しながら業務処理を進める方式が多くシステムで採用されている。この方式により、ビジネスロジックをアプリケーションから外出しにすることができ、業務管理者が直接業務ルールを把握・修正することが容易となり、業務アプリケーションの開発コストと運用コストの低減が実現できる。しかし、既存の方式では一定の形式でスプレッドシートを記述するか、スプレッドシートで記述しきれない場合は独自のルール言語でルールを記述する必要があった。

文献 1)では、自由な形式で記述したスプレッドシートを Java のライブラリで読み取ることで、幅広い用途に利用でき、さらに厳密な財務計算が必要な業務にも適用できる業務アプリケーションの実装方法を提案した。この方式により、オフラインで利用していた既存のスプレッドシートを利用してオンラインシステムの開発が行えるようになった。しかし、この方式では、画面から入力したデータをスプレッドシートのどのセルに代入すべきかを別途定義する必要があった。また、計算を実行する機能を複数のアプリケーションから利用できるように、独立した Web サービスとする場合は、アプリから Web サービスに送信する項目名と、Web サービスが受信するときの項目名のマッピングを定義する必要があり、開発工数を削減する上での課題となっ

いた。

そこで、本研究ではスプレッドシートに付加情報（アンテーション）を記述することで、セルに対応する項目名を少ない工数で指定できる方式を提案する。今回は、広く利用されている Microsoft Excel のファイル形式に対応しているという理由からスプレッドシートのエミュレートに Apache の POI ライブラリ (<http://poi.apache.org/>)を採用した。以下の第 2 節で本稿の提案するシステムの構成とスプレッドシートの記述例を示す。第 3 節で本稿の解決する課題を示す。第 4 節で付加情報の内容とその処理方法を示す。

### 2. システム構成

以下に本稿で提案するシステムの構成を示す。

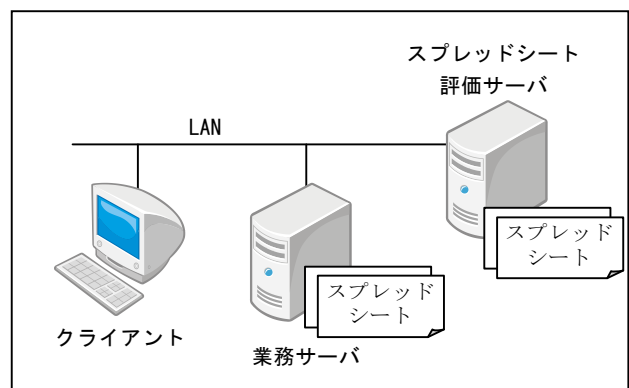


図 1 システム構成

本システムでは、画面の送信やユーザからの入力情報の収集は業務サーバが行う。Web の画面で使用する HTML はスプレッドシートのレイアウトを解析して自動生成する。スプレッドシートに記述されている計算などのロジックの処理は、業務サーバからスプレッドシート評価サーバに実行を要求する。スプレッドシート評価サーバは、Web サー

<sup>†1</sup> 三菱電機株式会社情報技術総合研究所  
Information Technology R&D Center, Mitsubishi Electric Corporation

<sup>†2</sup> 三菱電機インフォメーションシステムズ株式会社  
Mitsubishi Electric Information Systems Corporation

ビスを提供しており、指定されたスプレッドシートの評価を行いその結果を返す。スプレッドシート評価サーバは、複数のスプレッドシートを保持しており、異なる業務や複数の世代にも対応できる。

以下にお釣りを計算のスプレッドシートの例を示す。

	A	B
1	項目	値
2	価格	¥500
3	消費税	=B2*(1+B3) 0.05
4	税込価格	¥525
5	お預かり	= B5-B4 ¥1,000
6	お釣り	¥475

図 2 スプレッドシートの例

このスプレッドシートは品物の「価格」と「お預かり」金額を入力すると、お釣りを出力するシートの例を示している。B4 セルには「税込価格」を計算するための計算式「=B2\*(1+B3)」が書かれており、B6 セルには「お釣り」を計算するための計算式「= B5-B4」が書かれている。そして、価格として¥500、お預かり金額として¥1000 を代入すると計算結果として¥475 が表示される。

Apache の POI ライブラリは、上記のような Excel ファイルで記述されたスプレッドシートをメモリに読み込み、メモリ上でスプレッドシートをエミュレートすることができる。例として、POI ライブラリでお釣りを計算する処理の流れを以下に示す。

- (1) スプレッドシートをメモリに読み込む
- (2) 価格のセル「B2」に 500 を代入
- (3) お預かりのセル「B5」に 1000 を代入
- (4) B6 セルの値の評価を POI ライブラリに要求
- (5) POI ライブラリは、B6 に記述されている計算式に従い B5 セルと B4 セルを参照
- (6) B4 セルの計算式に従い、B2 と B3 セルを参照
- (7) 計算「500\*(1+0.05)」を実行して 525 を得る
- (8) 計算「100-525」を実行して 475 を得る

このエミュレーション機能により、自由な形式のスプレッドシートの利用が可能となった。一方、POI ライブラリでは内部で二進数を使用して数値計算を実行しており、厳密な財務計算に対応できないという課題があった。そこで、文献 1)では新たに「十進数演算機能」を開発して、それを適用して POI ライブラリを拡張し、誤差の問題を解決できることを示した。残る課題として以下がある。

### 3. 課題

#### 3.1 項目マッピングの課題

スプレッドシート評価サーバは、業務サーバから「価格」と「お預かり」と共に計算リクエストを受け取る。スプレッドシート評価サーバは、「お釣り」の計算をするために、

「価格」と「お預かり」をどのセルに代入すれば計算できるか、スプレッドシート評価サーバは知っている必要がある。

以下にお釣り計算リクエストの例を示す。

```
<?xml version="1.0" ?>
<env:Envelope ...
  <env:Body>
    <item name="価格" value="500" />
    <item name="お預かり" value="1000" />
  </env:Body>
</env:Envelope>
```

図 2 お釣り計算リクエスト

しかし、どのセルに設定すべきかをスプレッドシートから読み取ることにはできない。同様に「お釣り」がどのセルに出力されるかを、スプレッドシートから読み取るとは難しい。

そこで、文献 1)の方法では、各項目がどのセルに対応するかを示すマッピング定義を記述していた。また、値を代入するときに適切な型で設定する必要があるため型の情報も同時に記述していた。

以下にマッピング定義ファイルの例を示す。

```
<define>
  <mapping name="価格" cell="B2" type="number"/>
  <mapping name="お預かり" cell="B5" type="number"/>
  <mapping name="お釣り" cell="B6" />
</define>
```

図 3 マッピング定義ファイルの例

スプレッドシート評価サーバはこの定義を参照して、「価格」に対応するセルが「B2」であることを確認し、「B2」に値を代入する。その際、型として number (数値) が指定されていることから文字列を数値に変換してから代入する。このマッピング定義ファイルを利用する方式には、以下の課題がある。

- A) ファイルの作成は別途行う必要があり工数が掛かる
- B) 項目名が変更された場合はマッピング定義ファイルも修正が必要
- C) スプレッドシートが拡張されて、セルの位置が移動した場合はマッピング定義ファイルも修正が必要

#### 3.2 HTML 変換時の課題

本システムでは、画面の HTML をスプレッドシートのレイアウトを解析して自動生成する。例として「消費税」などの文字列は font タグに変換し、罫線で囲まれている矩形のエリアは、table タグに変換する。さらに、「B2」の価格を入力する欄は「input」タグに変換する必要がある。しかし、スプレッドシートのみから「B2」が入力欄であることを判別することは難しい。また、HTML ではユーザが入力した値を送信するためのボタンが必要になるが、元のスプレッドシートにはボタンは存在しない

ため、ボタンをどこに配置するかなどの定義を記述する必要があるのである。

HTML 変換時の課題を以下に記述する。

- D) 入力欄のセルを定義する
- E) 「送信ボタン」や「クリアボタン」を配置するための指定を与える

#### 4. 解決策

マッピング定義ファイルを独立したファイルとして定義するのではなく、スプレッドシートに付加情報を記述してその情報からマッピング情報を取得する方式が考えられる。以下に付加情報を追記した例を示す。

	A	B
1		値
2		¥500
3		0.05
4		¥525
5		¥1,000
6		¥475

`@Set(name="price")`

`@Set(name="amount")`

`@Get(name="change")`

`@Button (type="submit" value="送信")`

図 4 付加情報の例

上の例では、セル「B2」に「@Set(name="price")」という付加情報を付けており、「@Set」は代入するためのセルであることを表し、引数の name="price" は項目名が「price」であることを表している。セル「B6」の「@Get(name="change")」では、「@Get」によりこのセルが値を出力するセルであり、引数の name="change" により項目名が「change」であることを表している。また、セル自身が型の情報を持っており、セルの情報から型を求めることができる。以上により、マッピング定義ファイルで記述していた、「項目名」「セルの位置」「型」の情報がスプレッドシートから取得できるようになった。

HTML に変換するときの課題であった、入力用セルの区別も「@Set」から判別できるようになった。さらに、以下のように項目名を「input」タグの name 属性の値として指定することで、ブラウザから受け取る項目名が、計算サービスの項目名と一致するようになる。

```
<input type="text" name="price" />
```

さらに、「A6」は「@Button (type="submit", value="送信")」という付加情報を記述することで、この位置に「送信ボタン」を表示するもので、HTML に変換するとき submit タイプの「input」タグに変換する。

以上より各課題がどのように変換されたかを以下に記述する

- A) マッピングファイルはスプレッドシートから自動生成できるようになり工数が削減された

- B) 項目名は「@Set」で指定されている項目名が使用されるため修正は不要
- C) スプレッドシートが拡張されて、セルの位置が移動しても、付加情報がセルに記述されているため、位置がずれることはない
- D) 「@Set」により入力欄の判別が可能
- E) 「@Button」により「送信ボタン」や「クリアボタン」を配置

以下にアノテーション一覧を示す。

表 1 アノテーション一覧

No	名前	説明
1	@Set	入力用セルを指定する。テキストエリアとプルダウンのセルを指定する
2	@Get	出力用セルを指定する。計算結果や判定結果を出力するセルを指定する
3	@Button	サブミットボタンやクリアボタンを表示するセルを指定する

以下に上のスプレッドシートを HTML に変換してブラウザで表示した結果を示す。

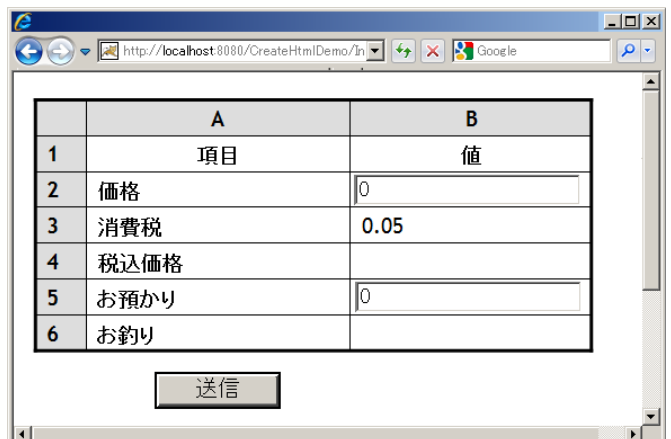


図 6 HTML に自動変換した例

#### 5. おわりに

本稿では、Web アプリケーションと Web サービスを自動的に連携させる方式について論じた。従来方式との違いは、マッピングファイルの代わりにスプレッドシートに付加情報を記述するのみで連携を可能とし、スプレッドシートのフォーマットが変更された場合でも連携処理は影響を受けないようにした点である。さらに、HTML に変換する際に必要となる情報も同じ形式で与えることができることを示した。今後はこの方式を実システムに適用してその有効性の評価を行う。

#### 参考文献

- 1) 小池賢一, 菅野幹人: "スプレッドシートを利用した業務アプリケーションの実装", 情報処理学 FIT 公演論文集, 2011.
- 2) 川口正高, 塩尻綾子, 浅見可津志, 原田雅史, 佐藤啓紀: オープン環境のシステム構築を高品质・短納期で実現する Web システム開発標準 "MIWESTA" 三菱電機技報 81(7), 489-492, 2007-07