

マッシュアップを使用したサービスの図示による保守性の向上
Improvement of maintainability by showing service using mashup

明治大学大学院理工学研究科基礎理工学専攻情報科学系
ソフトウェア工学研究室

Software Engineering Laboratory, Graduate School of Science and Technology,
Meiji University

小場 隆行 中所 武司
Takayuki Koba Takeshi Chusho

1. はじめに

近年、Web サービスにおけるAPI(Application Programming Interface)の公開に伴い、それらを組み合わせて使用マッシュアップ(Mashup)と呼ばれるWebサービス連携技術が普及しつつある。

本論文では、マッシュアップサービスの更なる増加を見越し、サービスの構成を図示化することによる保守効率の向上の手法について述べる。

2. マッシュアップサービスの保守上の問題点

マッシュアップとは、複数のWebサービスAPIを組み合わせることによって、あたかも一つのWebサービスのように振る舞わせるWebサービス連携技術のことである。これによって、従来よりも短期間に品質の良いものを作成することができる。

しかしながら、複数のWebサービスを連携するが故に、仕様変更など保守業務を行う際には、使用するWebサービスの各々の仕様や通信方式、提供している情報のデータ構造、APIの仕様を把握する必要がある。特に、第三者が保守業務を行う際には、これらの理解に手間取ることが予想される。

以上のことから、第三者がサービスの構造の概要を把握する手法として、マッシュ

アップを利用したサービスの図示化を提案する。

3. 例題アプリケーション

本研究の例題アプリケーションとして、Google map[1]とぐるなびWebサービス[2]を組み合わせた飲み会マップ[3]を使用する。

更に両サービスの中継役として、ぐるなびWebサービスより取得した情報の形式を変換するためのサービスとしてXML2JSON[4]を使用する。

このアプリケーションは、地図上の位置情報(経度・緯度)を検索キーとして、検索位置近辺のお食事処を地図上にヒット件数分のマーカーとして表示するというものである。一つのマーカーは一店舗に対応しており、クリックすることで営業時間などの詳細情報を閲覧することができる。

本アプリケーションの実行画面は以下の図1に示す。図1は小田急線登戸駅を中心として店舗情報を検索した際の出力結果であり、ある店舗の詳細情報を出力した画面である。

このような地図機能をシステムの中心としたマッシュアップのことをマップマッシュアップと言い、現在行われているマッシュアップの中で主流とされているタイプのものである。

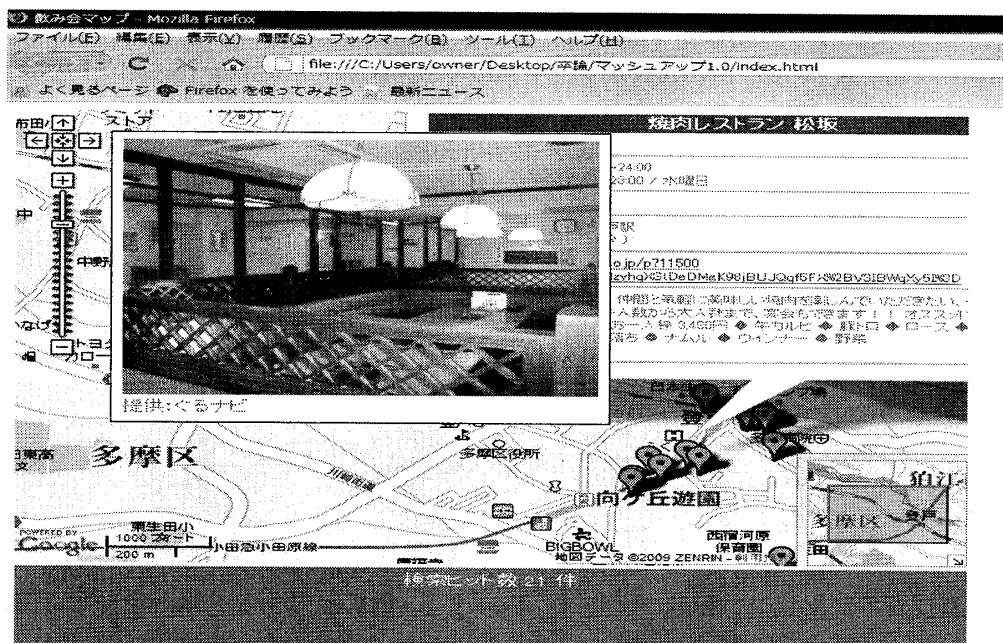


図1：例題アプリケーションの実行画面

4. マッシュアップを使用したサービスの図示化

2.で述べたように、マッシュアップを使用したサービスの保守には使用するWebサービスの仕様や通信方式、提供している情報のデータ構造、APIの仕様を把握していないとまらない。

使用するWebサービスの仕様とは、そのサービスを使用する上での規約に準ずるものやそのサービスで使用している言語に準ずる規則のことである。

通信方式とは、Webサービスとどのような手法で通信を行うかである。マッシュアップで利用される主な通信方式はREST(Representational State Transfer)とSOAP(Simple Object Access Protocol)である。

情報のデータ構造とは、Webサービスが返信するデータの構造のことである。XML(Extensible Markup Language)形式を使用した階層構造のものが多いが、Google mapのように独自に定義したクラスのオブジェクトとしてデータを返信する

など様々である。また、クロスドメイン問題の観点から例題アプリケーションではXML形式からJSON(JavaScript Object Notation)というデータ形式に変換している。

APIの仕様とは、クエリと呼ばれるWebサービスへのリクエストの記述方法とそれによってどのような情報が返送されるのかである。これらはWebサービスごとに定義されており、Webサービスの提供元のホームページにて閲覧することができる。

保守者はこれらの情報を踏まえた上で保守を行わないとまらないため、対象となるマッシュアップを使用したサービスの概要を知る手段が必要であると言える。その手段としてサービスの図示化を提案する。

サービスの図示化のために以下の記法を導入した。

- ① ファイル、Webサービス：二重線の四角の中に名前を記述。
- ② 関数：一重の四角の中に関数名を記述。
- ③ 独自のクラスを提供するサービス

- のオブジェクト:一重の四角で表現。
- ④ 関数呼び出し:呼び出し元を根とした単一方向の矢印で表現。(→)
 - ⑤ Web サービスへのリクエスト:単一方向の二重線矢印で表現。(⇒)
 - ⑥ Web サービスからのレスポンス:単一方向の破線矢印で表現。
 - ⑦ 通信方式:リクエストを表現している→の上に記入。
 - ⑧ データ構造:矢印の下にデータの源泉と形式変換が伴う場所のみ記入。
- これらを3.で述べた例題アプリケーションに適応した結果を図2に示す。
図3中のmapがGoogle mapから受け取

ったオブジェクトであり、その中には各種設定を行うことができるメソッドが含まれている。地図上のクリックなどのイベント処理は関数 `setEvent` で処理を行う。

ここで算出された位置情報を関数 `getGurunamiUrl` にてクエリとし、ぐるなびWebサービスと通信する。通信は、データ形式の変更を行う `XML2JSON` を経由して行う。

ぐるなびから受け取った情報は関数 `setMarkers` によってマーカーの内容として処理する。最終的に、下請け関数である関数 `createGMarker` にて地図上のマーカーとして設置される。

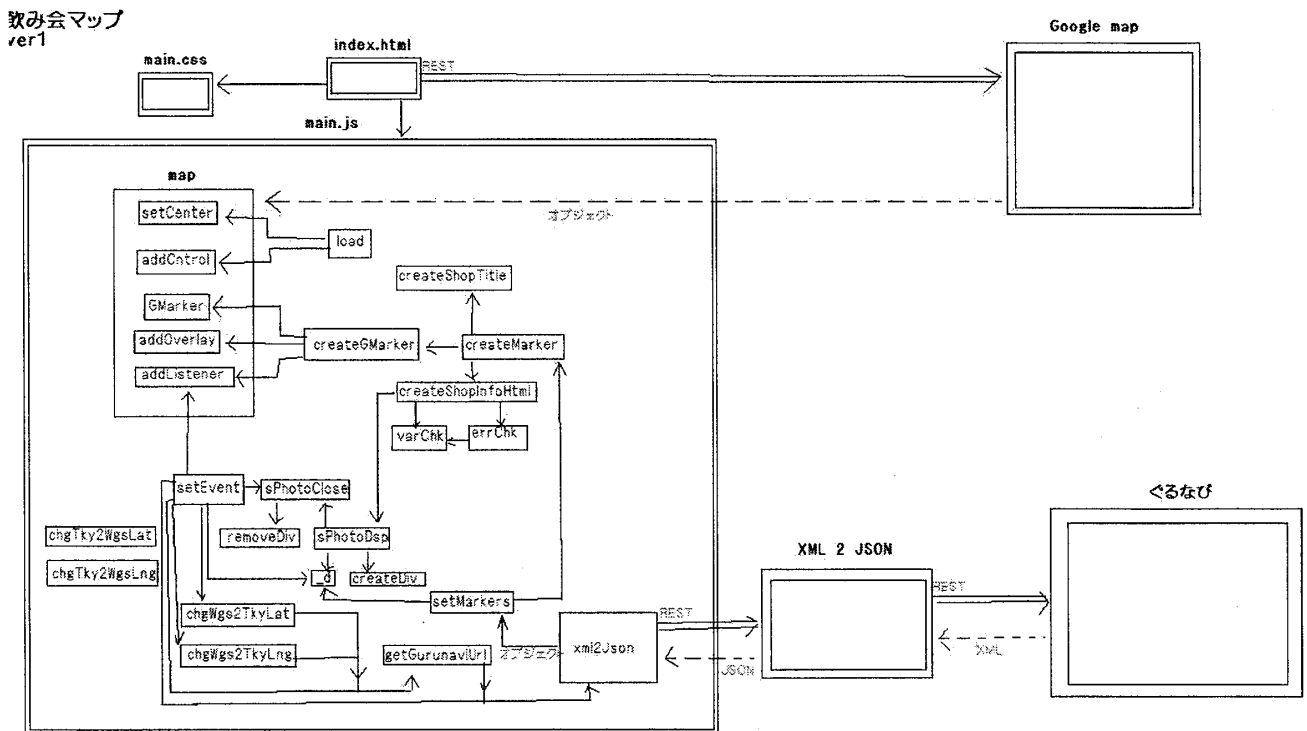


図2: 例題アプリケーションの図示化

5. 仕様変更の例

本章では例題アプリケーションで利用したぐるなびWebサービスをそれと類似の食事処検索サービスであるホットペッパーWebサービス[5]へ変更する場合を考える。

ぐるなびWebサービスとホットペッパーWebサービスは通信方式(REST)、データ形式(XML形式)共に同様の形式なため、比較的可変しやすいサービスであると言える。

ここで、ぐるなびWebサービスを変更し

た際に影響を受ける部分(関数 `setEvent` から関数 `sPhotoDsp` まで)について考える。これらの関数群は変更による影響を受けうる最大範囲であり、ぐるなび Web サービスの変更による変更部分はこの範囲内と言える。

このような関数群は、変更箇所の前後の関数や API を芋づる式に見ていき、仕様変更の影響の有無を判定して導き出すことが

可能である。

実際に、この仕様変更の影響を受ける部分を着色したものを図3に示す。図よりぐるなび Web サービスを中心に変更が必要であることがわかる。また、上述した影響を受ける範囲内で仕様変更されていることがわかる。

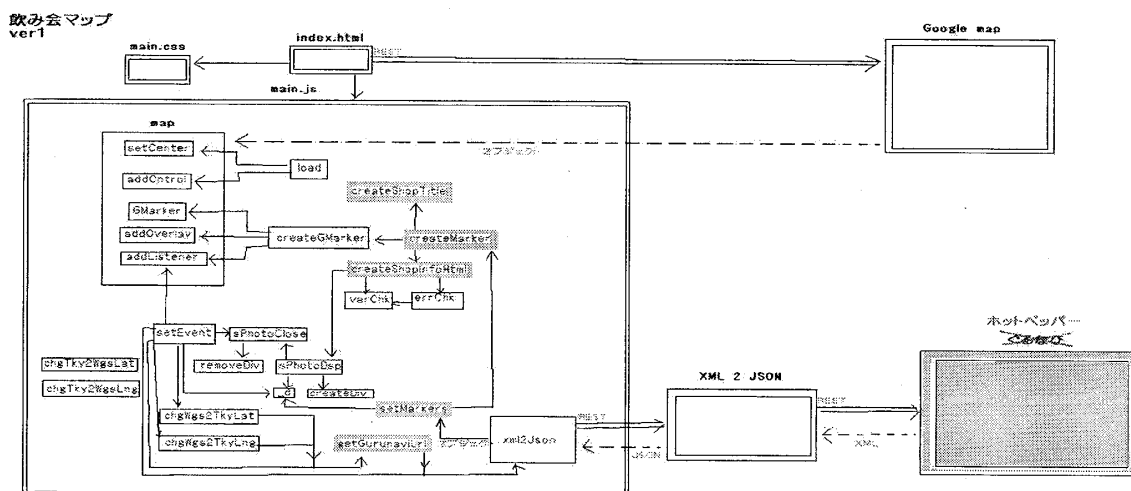


図3：ホットペッパーWeb サービスに変更する上で変更が必要な部分

6. おわりに

本論文では、マッシュアップを利用したサービスの保守効率の向上を目的とした図示記法について述べた。保守や仕様変更の対象となるシステムの構成を図から把握することによって、作業効率を向上させることができる。

また、発案したシステムを図示化した図に加え、関数の機能などを述べた関数表や使用している Web サービス、API の仕様書などを加えることによって、作業効率は更に改善されるであろう。

今後の課題として、例題アプリケーションを代表とするマップマッシュアップ以外のタイプのマッシュアップに対する対応

実験、検証を行う必要があるであろう。

参考文献

- [1] 『Google map』
<http://maps.google.co.jp/>
- [2] 『ぐるなび Web サービス』
<http://api.gnavi.co.jp/api/service.htm>
- [3] 本田正純：著 『マッシュアップかんたん AtoZ』 C&R 研究所：発行所
2007年10月1日：初版発行
- [4] 『XML2JSON』
<http://app.drk7.jp/xml2json/>
- [5] 『ホットペッパーWeb サービス』
<http://api.hotpepper.jp/>