

位置情報を含むライフログの可視化サービス開発支援フレームワーク

高橋 昂平[†] 下條 彰[†] 榎本 真佑[†] 中村 匡秀[†]

[†] 神戸大学 〒 657-8501 神戸市灘区六甲台町 1-1

E-mail: †{koupe,shimojo}@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし 本稿では、地図を用いたライフログサービスを容易に開発するためのアプリケーションフレームワーク MashMap フレームワークを提案する。提案フレームワークでは、様々な形式の位置情報付きライフログを標準データモデル (LLCDM 形式) に変換して DB に蓄積する。開発者は、必要なデータを DB から選別するためのフィルタと、そのデータの表示方法を作成して、データソースを定義する。次に、データソースを一つあるいは複数選択して、MashMap を作成する。MashMap とは、データソースを指定された表示方法で同一地図上に重ねて表示 (マッシュアップ) するオブジェクトである。MashMap は最終的に MashMap Renderer によって Google Map 上に可視化される。ケーススタディとして、旅行ログ振り返りサービスと研究会開催地マップを作成し、提案手法の有効性を示す。キーワード ライフログ, マッシュアップ, 位置情報, API, フレームワーク, 地図作成

An Application Framework for Developing Visualization Services of Location based Lifelog

Kohei TAKAHASHI[†], Akira SHIMOJO[†], Shinsuke MATSUMOTO[†], and Masahide NAKAMURA[†]

[†] Kobe University Rokkoudaityou 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: †{koupe,shimojo}@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

Abstract This paper presents an application framework, called *MashMap framework*, which facilitates development of location-based lifelog services. The proposed framework imports various types of location log data, and stores them in a database with the Life Log Common Data Model (LLCDM). A developer first creates a data source consisting of a filter and a display format. The filter extracts necessary data from the database, while the display format defines how the data is shown on the map. The developer then defines a MashMap by choosing a single or multiple data sources. The MashMap is an object, which aggregates several data sources on a single map, shown in the designated display format. The object is finally visualized on a Google Map by MashMap renderer. We conduct a case study developing a travel log review service and a conference history map using MashMap framework.

Key words lifelog, mashup, location data, API, framework, map creation

1. はじめに

人間の日常生活における行動をデジタルデータとして記録するライフログが注目を集めている。Web 上にはいくつものサービスが存在し、様々な種類のライフログを Web 上で記録し共有できる。最近では、スマートフォンや安価な GPS ロガーが広く普及し、位置情報が付与されたライフログ (位置付きライフログと呼ぶ) を比較的容易に記録できるようになってきている。代表的なサービスとして、移動経路を記録する GARMIN [1] や、ある時刻にある地点いたことを「チェックイ

ン」する foursquare [2] などが存在する。さらに、Twitter [3] や Facebook [4] のような従来からあるライフログサービスにも、位置情報を付加できる機能が備わってきている。

位置付きライフログの特徴は、いつ、どこで、何があったかを振り返ることができる点にある。「どこで」をわかりやすく振り返るために、ほぼ全てのサービスが地図を用いた位置情報の可視化を行っている。こうした位置付きライフログの可視化は、これまで各サービス提供者がそれぞれ独自の方法で開発してきた。しかしながら、ライフログの時系列データから位置情報を抜き出し、何らかの方法で地図上に描画するという処理は、

特定のサービスに依らない共通的な処理である．さらに位置情報は，特定の種類のライフログに依らない汎用的なデータである [5]．したがって，様々な種類の位置付きライフログを地図上に表現する汎用的な枠組みがあれば，新たなサービスを迅速かつ効率的に開発できる．

そこで本研究では，様々な種類の位置付きライフログデータを地図上に可視化するための汎用的なアプリケーションフレームワーク MashMap フレームワークを提案する．MashMap フレームワークは，様々な形式の位置付きライフログを，我々の研究グループで提案しているライフログ標準データモデル (LLCDM) 形式 [5] [6] に変換してデータベースに蓄積する．

アプリケーションの開発者は，必要なデータを DB から選別するためのフィルタと，そのデータの表示方法を定義して，データソースを作成する．次に，定義したデータソースを一つあるいは複数選択して，MashMap を定義する．MashMap とは，データソースを指定された表示方法で同一地図上に重ねて表示 (マッシュアップ) するデータオブジェクトである．MashMap オブジェクトは最終的に MashMap Renderer によって Google Map 上に可視化される．提案フレームワークを用いることで，開発者は地図を用いた様々なライフログサービスを，迅速かつ容易に開発することが可能となる．

また本研究では，MashMap フレームワークを JavaServletAPI を用いて Google AppEngine 上に実装した．またケーススタディとして，MashMap ビューア，旅行ログ，研究会開催マップを作成した．その結果，提案手法によってサービス開発工数を効率化できる見通しを得た．

2. 準備

2.1 位置付きライフログ

位置情報が含まれたライフログを，本稿では位置付きライフログと呼ぶ．位置付きライフログはライフログの一種であるため，基本的には人間の何らかの行動やイベントの記録を表している．各記録には，その行動が発生した日付や時刻，関係するユーザがひもづけられる．加えて位置付きライフログは，その行動が発生した位置 (場所) の情報も含んでいる．具体的には，その場所の [緯度，経度，高度] の組，もしくは住所で表現される．位置情報は記録する行動の内容に強く依存しない汎用的なデータであり [5]，写真やつぶやき，運動記録など様々なライフログに付加することができる．

近年では GPS を備えたスマートフォンや安価な GPS ロガーが普及してきており，位置付きライフログを比較的容易に記録できるようになってきている．位置付きライフログの，誰が (Who)，いつ (When)，どこで (Where)，何を (What) かという情報は，人間の行動を詳細に性質づける情報である．したがって，個人行動の振り返りにとどまらず，健康，エンターテインメント，交通，流通，農業，防災など様々な分野でのサービス応用が期待される．

2.2 位置付きライフログを用いたサービス例

位置付きライフログを用いたサービスをいくつか紹介する．図 1(a) は，ウェザーニューズのウェザーリポートチャンネル



図 1 位置付きライフログを用いたサービスの例

ル [7] の画面である．このサービスでは，ユーザが携帯電話で各地の天気をピンポイントでレポートするサービスである．ユーザは，自分がいる場所の天気情報に加えて，写真やコメントなどを記録し，位置情報と共にログとしてサーバに送信する．各地のユーザから送信されたログは，同一マップ上に可視化される．サービスの利用者は，マップ上のマーカーをクリックすることで，その地点のピンポイントの天気を写真やコメント付きで見ることができる．

図 1(b) は，みんなでつくる放射線量マップ [8] の画面である．福島原発事故以来開発されたこのサービスは，参加者がそれぞれ自ら放射線量を測定し，測定地点の場所情報と共にサーバに送信する．各地で測定された放射線量は，一つの Google Map 上に集約されて描画され，広範囲の放射線量マップが作成される．地図上のマーカーをクリックすると，放射線の測定量，測定日時，測定地点，測定機器などの情報が表示される．

これらの他にも，GARMIN [1] や foursquare [2]，Twitter [3]，Facebook [4] など，様々な位置付きライフログサービスが存在する．図 1 で見たように，位置付きライフログを用いたサービスでは，表示するデータの内容や表示形式はそれぞれ異なるものの，地図上にデータを可視化することは共通している．

2.3 先行研究：標準データモデル (LLCDM) とマッシュアップ API (LLAPI)

ライフログの高度かつ柔軟なマッシュアップを支援するため，我々は先行研究 [5] [6] において，ライフログのための標準データモデル (Life Log Common Data Model, LLCDM) とマッシュアップ API (Life Log API, LLAPI) を提案している．

表 1 に LLCDM のデータスキーマを示す．LLCDM は 5W1H の観点からライフログが備えるべきデータ項目を整理し，特定のサービス・アプリケーションに依存しないデータストアを定義する．LLCDM の特徴は，日付や時刻，ユーザや場所といった，おおよそどのライフログにも付与される情報を標準化することである．また，アプリケーションに依存するログの内容に関しては自ら解釈せず，外部のスキーマに任せるといった構造をとっている．

LLAPI は LLCDM 形式に変換されたライフログデータにアクセスするための標準的なインターフェースを定義する．LLCDM のデータ項目に関するクエリを入力とし，合致するライフログデータを LLCDM 形式の配列で返す．

表 1 ライフログ標準データモデル (LLCDM) のスキーマ

観点	データ項目	説明
WHEN	<date>	日付
	<time>	時刻
WHO	<user>	ユーザ
	<party>	共に行動したユーザ
	<object>	対象ユーザ
WHERE	<location>	場所
HOW	<application>	ライフログサービスを 実現するアプリケーション
	<device>	ログ記録に用いたデバイス
WHAT	<content>	ログの内容
	<ref_schema>	外部スキーマへの参照
WHY	-----	必要なら<content>内へ

API 書式

```
getLifeLog(s_date, e_date, s_time, e_time, user,
           location, application, device, select)
```

パラメータ

s_date: 検索開始日, e_date: 検索終了日,
s_time: 検索開始時刻, e_time: 検索終了時刻,
user: ユーザクエリ, location: 場所クエリ,
application: アプリケーションクエリ,
device: デバイスクエリ, select: 抽出データ項目

LLAPI は Web サービスとして公開されており, 様々なプラットフォームから呼び出しが可能となっている。

2.4 現状の課題

2.2 節で触れたように, 様々な位置付きライフログサービスでは, データを地図上に可視化する処理を行っている。この共通した処理は, 現在各サービス提供者が独自に開発しており, 再利用されていない。また, 位置情報を用いた機能も各サービスごとにばらばらで, 位置付きライフログを必ずしもフル活用できていない。例えば Twitter には「自分がつぶやきを投稿したところのある全ての場所を表示する」という機能はない。

様々な形式の位置付きライフログを地図上に可視化する汎用的な枠組みがあれば, 新たなサービスを迅速に開発することが可能になると我々は考える。さらに, 複数の種類の位置付きライフログを, 同一地図上で重ねて表示 (マッシュアップ) することができれば, ユーザは複数のデータを組み合わせ, より付加価値の高い地図を作ることができる。

本研究ではこれらの課題に対処するアプリケーションフレームワークの開発を目指す。

3. 位置情報を含むライフログの可視化サービス開発支援フレームワーク

3.1 MashMap フレームワーク

本研究では, 様々な形式の位置付きライフログを地図上に可視化するためのアプリケーションフレームワーク MashMap フレームワークを提案する。MashMap とは, 様々な位置付きログをマッシュアップ (Mashup) して地図 (Map) に表示するという意味を含めた造語である。フレームワークにおいて開発者

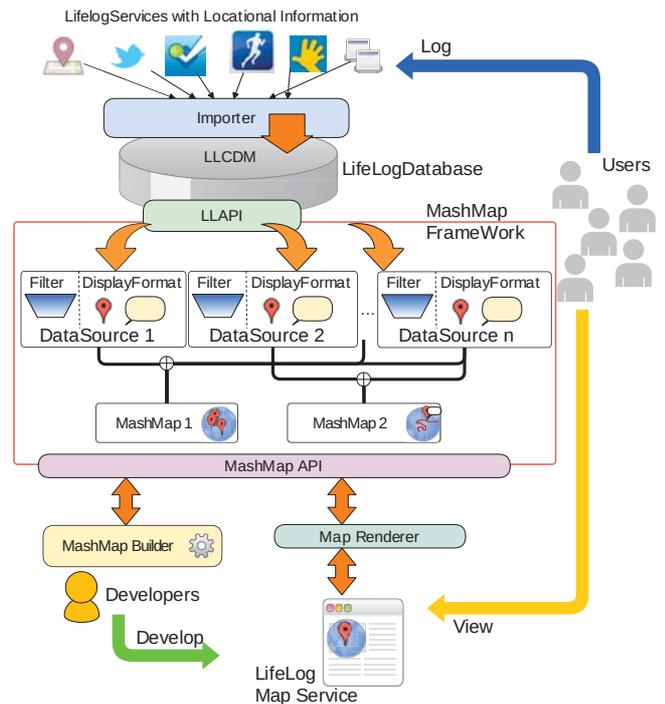


図 2 MashMap フレームワークの全体図

は, 位置付きライフログから自分の必要なデータを選んでデータの源泉 (データソース, data source) とし, 複数のデータソースをマッシュアップした地図, すなわち MashMap を作成する。

図 2 にフレームワークの全体像を示す。図の最上部は, ユーザが様々なサービスで記録した位置付きライフログを表している。これら様々な形式の位置付きライフログデータを, 提案フレームワークのデータベース (DB) へ随時インポートする。インポートの際, 2.3 節で述べた LLCDM 形式に適宜変換する。

アプリケーションの開発者は, DB に蓄積されたライフログデータから自分が必要なものを選別して, データソースを作成する。データソース作成においては, データの選別に用いるフィルタ (filter) と, 選別したデータをどのように地図上に表示するかを定義する表示形式 (display format) を指定する。

次に開発者は, MashMap オブジェクトを作成し, 作成済みのデータソースに接続する。異なるデータソースを複数組み合わせ合わせて接続することが可能である。こうして作成された MashMap オブジェクトは, MashMap Renderer によって地図形式に可視化される。地図上では, MashMap 内の複数のデータソースが, それぞれ指定された表示形式で描画され, 複数の位置付きライフログがマッシュアップされた地図ができる。

MashMap の作成やデータソースの設定などは全て MashMap API を通じて行われる。フレームワークのコアの部分 (図 2 中央四角部分) をクラウド上のサーバに配置して, MashMap フレームワークをサービスとして使う (Software as a Service, SaaS) ことも可能である。図 3 に提案フレームワークのクラス図を示す。各クラスについて以降で詳細に説明する。

3.2 LogData

前節で述べたとおり, MashMap フレームワークで利用する位置付きライフログは全て LLCDM 形式に標準化され, DB に

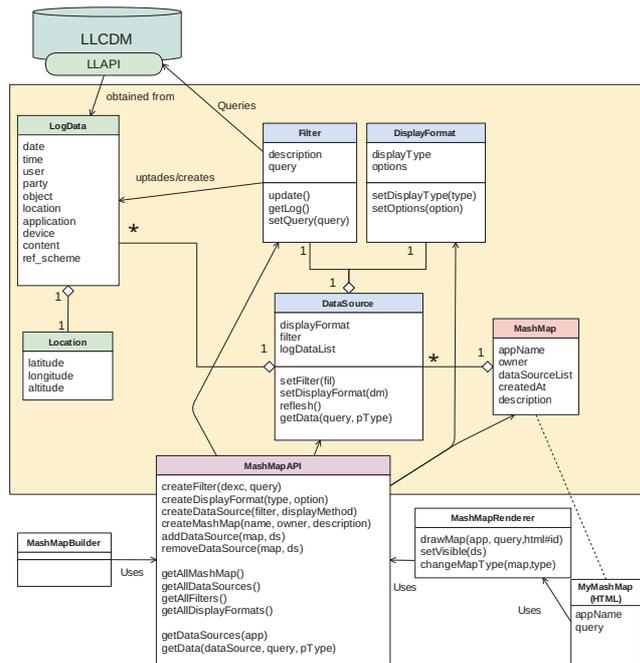


図 3 MashMap フレームワークのクラス図

蓄積される。フレームワークからは、2.3 節で述べた LLAPI の `getLifeLog()` を用いてデータを検索し取得する。フレームワーク内に取り込まれたデータの各レコードは、`LogData` クラスのオブジェクトに格納される。`LogData` クラスの属性は LLCDM 形式に準じている。例えば、「ユーザ koupe が 2011 年 9 月 11 日の 16 時 25 分に iPhone4 から foursquare にアクセスし、徳島県徳島市の眉山公園にチェックインした」という `LogData` オブジェクトは以下のようにシリアライズされる。

```
[2011-09-11, 16:25:00, "koupe", --, --, [34.066987, 134.532727, 256.7244873046875], foursquare, iPhone4, "...shout:ひたすら登る@眉山公園...",https://ja.foursquare.com/v/%E7%9...
```

3.3 Filter

`Filter` は DB から必要なデータを選別するためのクラスである。フレームワークを利用する開発者によって `Filter` オブジェクトが生成され、必要なデータを取り出すためのクエリがセットされる。`Filter` オブジェクトは指定されたクエリに基づき、LLAPI を実行し、必要なログを DB から取得、`LogData` オブジェクトとしてフレームワーク内に取り込む。`Filter` オブジェクトにセットされるクエリは、LLAPI の `getLifeLog()` のパラメータにしたがう。例えば、9 月 11 日中、ユーザ koupe が foursquare および GARMIN のそれぞれで記録したログを別々のデータ群として選別するには、以下の filter を生成する。

```
filter1= [description:"2011/09 徳島の foursquare",
  query:{2011-09-11, 2011-09-12, 00:00:00, 00:00:00,
  koupe, *, "foursquare", *, *)}]
filter2= [description:"2011/09 徳島の GARMIN",
  query:{2011-09-11, 2011-09-12, 00:00:00, 00:00:00,
  koupe, *, "GARMIN", *, *)}]
```

3.4 DisplayFormat

`DisplayFormat` クラスは、`Filter` オブジェクトにより選別されたデータ群を、地図上にどのように表示形式で描画するかを定義するオブジェクトを規定する。様々なデータを地図上に表示する時、そのデータ群ごとに適切な表示方法は異なってくる。例えば、GARMIN によって記録された軌跡であれば、位置座標を線で結ぶように表示したいし、Twitter によるつぶやきであれば、地図上にマーカーとして表示したい。さらに、同じ内容のデータ群であっても、組み合わせる別データとの兼ね合いによって、その表示方法を変更できれば、より効果的な視覚化が可能となる。各 `DisplayFormat` オブジェクトは、プロパティとして、位置データの表示タイプとそのオプションを持つ。表示タイプは、マーカー (Marker)、アイコン (Icon)、線 (Polyline)、吹き出し (InfoWindow) の 4 種類のいずれかの値をとる。オプションは、GoogleMaps JavaScriptAPI V3 の仕様に基づいて、JSON 形式で記述する。例えば、オブジェクトの種類にマーカーを指定し、影を表示しないという場合の `DisplayType` オブジェクトは

```
disp_marker = [Marker, "flat:true"]
```

また、オブジェクトの種類を線とし、線の色を赤色、太さを 2、不透明度を 100%とする場合は

```
disp_line = [Polyline, "strokeColor:#FF0000,
  strokeWeight:2, strokeOpacity:1"]
```

のような形になる。

3.5 DataSource

`DataSource` クラスは、開発者が必要とする位置付きライフログのデータ源泉を規定する。そのオブジェクトは、`Filter` と `DisplayFormat` それぞれへの参照を持つ形になっている。フィルタと表示形式がセットされたデータソースにおいて、データ取得メソッド `getData()` が実行されると、フィルタを用いて DB から必要なデータを選別して、`DataLog` オブジェクトの系列を得る。また、`getData()` にはクエリを指定して、選別したデータをさらにフィルタリングをかけることができる。これは、`MashMap` 上で表示する際に、取得したデータの絞る用途などに用いる。

取得されたデータは地図上で指定された表示形式でレンダリングされる。地図上へ表示されるライフログデータは、この `DataSource` オブジェクトの単位で扱われる。さらに、データソースごとに、表示・非表示を選択できる。これまでの例を用いて、徳島の foursquare のデータを、マーカーで表示する時のデータソースは以下ようになる。

```
ds_foursq = [filter:filter1, displayType:disp_marker,
  logDataList:[...]]
```

また、GARMIN のデータを線で表示するデータソースは

```
ds_garmin = [filter:filter2, displayType:disp_line,
  logDataList:[...]]
```

となる。

3.6 MashMap

`MashMap` クラスは、複数のデータソースを集約し、それらを重ね合わせた地図、`MashMap` オブジェクトを規定する。その

他に属性として、名前、開発者、開発した日付、説明などを含む。MashMap オブジェクトの例としては、例えば移動経路情報と、つぶやき、写真を組み合わせた「旅行ログマップ」や、全国各地の天気とコメントを組み合わせた「お天気マップ」などが考えられる。これまでの例を用いると「徳島旅行ログマップ」は以下のような MashMap オブジェクトとして作成できる。

```
["2011/09 徳島日帰り旅行ログ", "koupe", [ds_foursq, ds_garmin], 2012-01-03 07:05:12, "徳島へ日帰りツーリング. GARMIN と foursquare のログから作成"]
```

3.7 MashMapAPI

MashMapAPI は MashMap フレームワークの各種オブジェクトを操作するためのインターフェースを提供するファサードクラスである。Filter や DisplayFormat, MashMap の各種オブジェクトの生成、消去、更新を行う。また、データソースの一覧を取得したり、データソースからデータ系列を取得したりすることができる。新しく「徳島旅行ログマップ」を作成する場合の API の呼び出し系列を以下に示す。

```
/* MashMap 作成 */
tokushiima = createMashMap("徳島日帰り旅行
ログ", "koupe", "2011/09 の徳島日帰り旅行");
/* GARMIN のデータソースを作成 */
f1 = createFilter({user:koupe, application:GARMIN,
s_date:2011-09-11,...});
d1 = createDisplayFormat(Polyline, {strokeColor:
#FF0000, ...});
garmin = createDataSource(f1, d1);
/* foursquare のデータソース作成 */
f2 = createFilter({user:koupe, application:foursquare,
s_date:2011-09-11,...});
d2 = createDisplayFormat(Marker, {flat:true,...})
fsq = createDataSource(f2, d2);
/* MashMap にデータソースを登録 */
addDataSource(tokushima, garmin);
addDataSource(tokushima, fsq);
```

3.8 MashMapRenderer

MashMapRenderer は、MashMap を Google Map 上に可視化するための JavaScript ライブラリである。MashMap オブジェクトの参照 ID を渡すと、MashMapAPI を実行してデータソースを取得する。各データソースから位置付きライフログデータの系列を指定された表示形式で、地図上にレンダリングする。また、各データソース毎に地図上へのデータの表示・非表示を切り替える機能も持っている。また、地図の種類(市街地図、航空写真、地形地図など)を切り替えるための関数も備えている。

3.9 MashMapBuilder

MashMapBuilder は、MashMapAPI の開発者向け GUI フロントエンドである。MashMap をオブジェクト画面上の簡単な操作で作成する。画面上で Filter, DisplayFormat, DataSource をそれぞれ定義・選択し、新たな MashMap オブジェクトを簡単かつ迅速に作成することができる。また、作成された MashMap

表 2 実装に使用した技術

プラットフォーム	Google App Engine for Java
開発言語	Java, HTML, JavaScript
ライブラリ	Google Maps JavaScript API V3 jQuery1.7.1
CSS フレームワーク	Twitter Bootstrap v1.4.0
jQuery プラグイン	Twitter Bootstrap Dropdown Twitter Bootstrap Tabs DatePicker
動作確認ブラウザ	Google Chrome 16.0



図 4 MashMap ビューア

オブジェクトを可視化するための最小限のスケルトンを HTML ファイルとしてエクスポートする機能も備えている。開発者は、このスケルトンを適宜用途に合わせてカスタマイズし、位置付きライフログサービスを開発する。

3.10 実装

提案した MashMap フレームワークを実装した。表 2 に開発に使用した技術をまとめる。

4. ケーススタディ

ケーススタディとして、提案フレームワークを用いて、いくつかの実用サービスを作成した。

4.1 MashMap ビューア

MashMap フレームワーク上に作成・登録されている MashMap を簡単に閲覧、チェックできる Web アプリケーションを開発した。図 4 にスクリーンショットを示す。画面には、ユーザこうぺが作成した MashMap の一覧が表示されている。閲覧したい MashMap を選択すると、地図画面に移動する。地図画面では、各種データソースの表示・非表示を切り替えたり、年や月、日などでデータを絞り込んで表示したりできる。

MashMap ビューアは HTML, JavaScript, JavaServlet を用いて開発し、有効行数はトータルで 680 行であった。処理の大部分が提案フレームワークをそのまま利用しており、効率的な開発が実現できたと考えている。

4.2 旅行ログ

3 章の各節で用いた例題「徳島旅行ログマップ」を MashMap

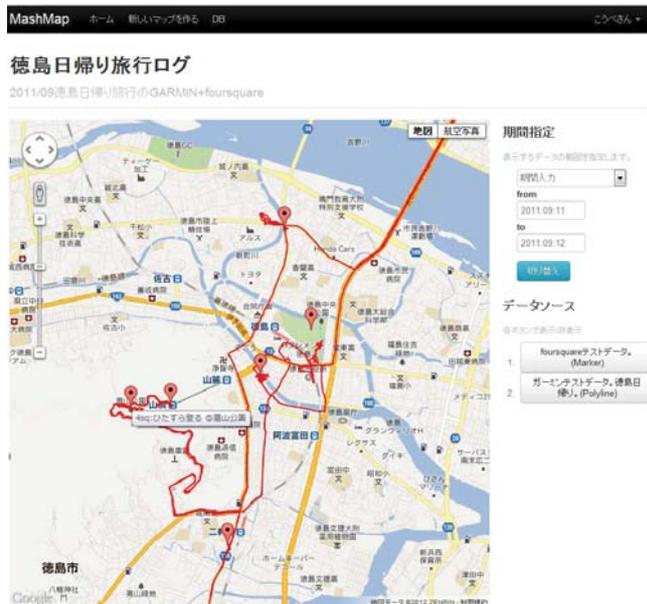


図 5 徳島日帰り旅行ログ

フレームワークにより作成した．図 5 に MashMap ビューアでの確認画面を示す．画面には，筆者が 2011 年の 9 月 11 日に行った徳島県徳島市への日帰り旅行の位置付きライフログを可視化している．地図中の赤線が GARMIN のデータソース，マーカーが foursquare のデータソースを表す．マーカーをマウスオーバーすることで，場所の名前とコメントが表示される．地図右側の期間指定により，表示するログの期間を絞り込むことができる．その下「データソース」は各データソースの表示 / 非表示を切り替えるボタンである．

4.3 研究会開催地マップ

別の例題として，LOIS 研究会（過去名称 OIS, OFS 研究会）の歴代の開催場所のログを MashMap として作成した例を図 6 に示す．この地図には，平成 11 年度 5 月から平成 23 年度 3 月までに行われたすべての研究会開催地のデータが関連付けられている．マーカーによりその開催地の位置が示され，マウスオーバーで開催日とその場所の名称が表示されるようになっている．

5. 関連研究

位置付きライフログを利用し，地図上で可視化を行うための研究は数多く行われている．文献 [9] では，防災情報マッシュアップシステム GDMS を構成し災害支援に役立てようとしている．文献 [10] では，地理位置情報と時系列情報を同時に表示し，その関係を実感的に理解するための時系列地理位置情報表示システムを試作している．文献 [11] では，植物生長予測モデルや，病害発生予測モデル，栽培管理支援システムなどの異なる農業用アプリケーションの結果や蓄積された情報を地図上で結びつける研究を行っている．

これらの既存研究では，それぞれの分野に特化した表示方法，データ構造を利用し，独自の方法で地図の作成を行っている．これに対し MashMap フレームワークは，特定のログ，特定の



図 6 OFS/OIS/LOIS 開催地マップ

アプリケーションに依存しない汎用的なフレームワークであり，表示形式やデータソースを柔軟に変更できるカスタマイズ性や，異種データの相互運用性に優れている．

6. おわりに

本稿では，位置付きライフログを地図に可視化するための MashMap フレームワークを提案した．提案フレームワークにより，地図を用いたライフログアプリケーションやサービスを迅速に開発することが可能となる．今後の課題としては，MashMap フレームワークを強化し，位置情報を持たないデータのマッシュアップや，地形による絞込み，データの集計・分析を支援する機能を実装していきたい．

謝辞 この研究は，科学技術研究費（基盤研究 B 23300009，若手研究 B 21700077，研究活動スタート支援 22800042），および，ひょうご科学技術協会の助成を受けて行われている．

文 献

- [1] GARMIN Connect. <http://connect.garmin.com>.
- [2] foursquare. <http://foursquare.com/>.
- [3] Twitter. <http://twitter.com/>.
- [4] Facebook Places. <http://facebook.com/places/>.
- [5] 中村匡秀，下條 彰，井垣 宏，“異なるライフログを集約するための標準データモデルの考察” 信学技報，vol.109，no.272，pp.35-40，2009．
- [6] 下條 彰，福田将之，井垣 宏，中村匡秀，“異なるライフログをマッシュアップするためのデータ変換・集約アクセス API の実装” 信学技報，vol.109，no.450，pp.85-90，2010．
- [7] ウェザーリポート Ch. <http://weathernews.jp>．
- [8] みんなでつくる放射線量マップ．<http://minnade-map.net/>．
- [9] 加藤孝明，小林三昭，佐藤尚秀，四柳照義，“防災情報マッシュアップシステムの社会への実装に向けて” 生産研究，vol.62，no.4，pp.377-380，2010．
- [10] 桑原一聖，服部 哲，速水治夫，“地図 API とタイムライン API のマッシュアップによる時系列地理位置情報表示システムの試作（地図の利用）” 情報処理学会研究報告．GN 研究会，vol.2009，no.33，pp.109-114，2009-03-11．
- [11] 田中 慶，平藤雅之，“農業モデルにおける web 地図サービスを利用した地図インタフェース” 農業情報研究，vol.18，no.2，pp.98-109，2009．