Vol.2009-DPS-140 No.1
2009/9/10

リアル指向型 Web サービスのモデル化とフレームワークの検討

泉森達也[†1]　小板隆浩[†2]　佐藤健哉[†1]

現在，GPS や加速度センサなど情報を利用できるモバイル機器が普及し，様々なセンサ情報を活用した Web サービスが台頭している．このような Web サービスをリアル指向型 Web サービスと定義する．リアル指向型 Web サービスを開発する上で，センサ情報を扱うことを想定していない既存のフレームワークを適用できない．そのため，開発コストが高いことが問題となる．

本研究では，既存フレームワークとの対比からリアル指向型 Web サービスの構築方法をモデル化する．更に，モデルに基づいたフレームワークを設計し，実際の Web サービスへの適用について検討する．

Modeling of Real-oriented Web Services and Designing of the Framework

Tatsuya Izumori,[†1] Takahiro Koita[†2] and Kenya Sato[†1]

Mobile devices using sensors such as GPS or acceleration sensor are widely used. Web services that use information obtained from many kinds of sensors are increasing. We define such Web service as real-oriented web service in this paper. Existing frameworks cannot be applied to real-oriented web services, because those frameworks are not considered to handle such sensor information.Therefore, the developing cost with existing frameworks is high and its cost have to be reduced.

In this paper. we investigate two existing frameworks, and propose a new model for real-oriented web services. Furthermore, we discuss the details of the proposed model and application scenario for actual web service.

1. はじめに

1.1 背　景

近年，モバイル機器のセンサ機能を用いて，位置情報や方位情報などを利用する Web サービスが台頭し始めている．こうした Web サービスが普及してきた背景として，GPS や加速度センサなどのセンサが標準で利用できる携帯電話などのモバイル機器の普及が挙げられる．また，Android や iPhone などの普及によって，センサ情報を利用したシステムやサービスの開発が可能になり，様々なアプリケーションがリリースされている．

1.2 リアル指向型 Web サービス

本研究では，モバイル機器のセンサから取得できる位置情報や方位情報などの情報を活用した Web サービスをリアル指向型 Web サービスと呼ぶ．代表的なリアル指向型 Web サービスは，2008 年 11 月に頓知・(トンチドット) 社から発表されたセカイカメラ[1)] である．セカイカメラは Apple 社が販売する携帯電話の iPhone 専用のアプリケーションであり，iPhone の GPS・加速度センサ・地磁気センサなどを利用してカメラ映像上に情報を表示することができる．モバイル機器のカメラ映像上にテキスト・画像・音声などを配置することができ，ナビゲーションや広告などを表示する新たなコミュニケーションサービスとしての利用が期待されている．このように，リアル指向型 Web サービスは，センサを用いて現実世界における様々な情報を取り入れることで，新しいコミュニケーションやサービスを実現できる Web サービスである．

これまで実装されてきた既存のリアル指向型 Web サービスは，開発におけるモデルが存在しないため，それぞれ独自の実装で実現されている．よって，開発者はモデル化されていない部分について設計する手間がかかるため，開発コストが大きい問題がある．また，リアル指向型 Web サービスの開発が増加しつつあり，開発コストを削減することが望まれている．そこで，本研究ではリアル指向型 Web サービスのモデル化を行い，開発コストを削減できるフレームワークを提案し，その検討を行う．

†1 同志社大学大学院工学研究科
Graduate School of Engineering, Doshisha University
†2 同志社大学理工学部
Faculty of Science and Engineering, Doshisha University

©2009 Information Processing Society of Japan

2. 現状の問題点

本章では，一般的な Web アプリケーションフレームワークと VCCM(View-Controller-Controller-Model) フレームワーク[4] を挙げ，それぞれのモデルについて述べる．その上で，現状のリアル指向型 Web サービスの開発における問題点を明らかにする．

図 1 に一般的な Web アプリケーションフレームワーク，図 2 に VCCM フレームワークのモデル構成の概要をそれぞれ示す．それぞれのフレームワークは MVC(Model-View-Controller) アーキテクチャ[2] に従って実装されている．の M・V・C 各要素は M が Model, V が View，C が Controller を表す．MVC アーキテクチャとは，ソフトウェアを Model, View, Controller にそれぞれ分割して設計・実装する技法・あるいは構造である．[⋆1]

2.1 Web アプリケーションフレームワーク

Web アプリケーションフレームワークは，Web サービスの開発コスト削減の手法としてとして開発者に広く採用されている．Web アプリケーションフレームワークの最も代表的なものとして，Ruby on Rails[3] が挙げられる．Web サービス開発者はフレームワーク側で定められたモデルに従ってコードを配置するだけで Web アプリケーションを開発することができる．Web アプリケーションフレームワークを用いた開発では，新たにアプリケーションの構成を設計する必要がないため，Web サービス開発の生産性を大きく高めている．Web アプリケーションフレームワークによる Web サービス開発のモデル構成を図 1 に示す．図中の Application は，Web サービスにリクエストを送るプログラムを指す．Web アプリケーションフレームワークにおいて想定する Application は Web ブラウザである．

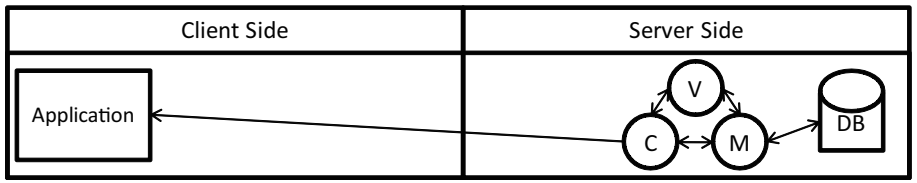

図 1 Web アプリケーションフレームワークのモデル構成

Web アプリケーションフレームワークは，クライアント側の Application から HTTP リクエストを受け，サーバ側にあるデータを処理して HTML や XML を返すタイプの Web サービスに対応している．このような Web サービスはクライアント側のデータについて複雑な処理を記述する必要がないため，Web アプリケーションフレームワークではクライアント側の処理をモデル化していない．そのため，クライアント側の複雑な処理を記述する必要があるリアル指向型 Web サービスの開発には対応しきれないという問題がある．

2.2 VCCM

VCCM フレームワークは，Web サービス開発のフレームワークの中でも，クライアント側の処理の記述に重きを置いたフレームワークである．VCCM フレームワークは，Application において Flash などのリッチな処理を記述できるプログラムから利用される Web サービスの開発に特化したフレームワークである．リッチな処理を記述できるクライアントに対応するため，MVC アーキテクチャの中の Controller をクライアント側の Controller とサーバ側の Controller に分けた点が特徴である．VCCM フレームワークにおけるサーバクライアント間の通信は XML で規定され，通信を意識することなく開発することができる．VCCM フレームワークによる Web サービス開発のモデル構成を図 2 に示す．

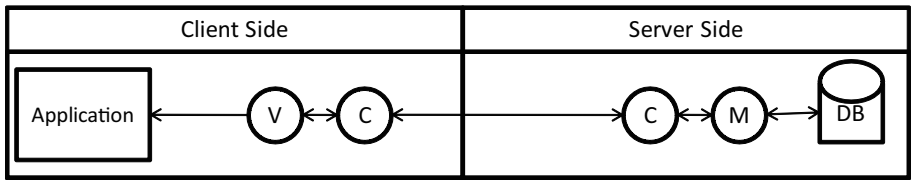

図 2 VCCM フレームワークのモデル構成

VCCM を用いることで，クライアント側に複雑な処理を記述することが可能になる．しかし，クライアント側に存在するデータを取得することは考慮されていないモデルであるため，センサからデータを取得する必要があるリアル指向型 Web サービスにそのまま適用することは難しい．

2.3 現状のリアル指向型 Web サービス

リアル指向型 Web サービスでは，センサからデータを取得する必要がある．しかし，既存のフレームワークでは，クライアント側のデータを取得するモデルが規定されておらず，センサからデータを取得する概念がない．また，クライアント側のデータを通信するための

⋆1 ただし，本研究で述べるモデルと MVC アーキテクチャにおける Model は異なるものを指す．

ⓒ2009 Information Processing Society of Japan

データ形式も規定されていない．更に，クライアント側のセンサ情報を通信したり，Viewへ受け渡したりするために，Controllerでそれら統合的に記述できる必要がある．例えば，VCCMでリアル指向型Webサービスを開発しようとする場合，開発者はセンサ情報を通信する際のデータ形式・センサ群とのアクセス方法・データ処理の統合方法を設計しなければならない．そのため，既存のフレームワークをリアル指向型Webサービスにそのまま適用することはできない．

3. 提 案 手 法

3.1 リアル指向型 Web サービスの必要要件

リアル指向型Webサービスのモデル化にあたって，クライアント側では以下の3点の要素を設計する必要がある．

- **センサ情報を通信する際のデータ形式**：クライアント・サーバ間で通信するセンサ情報の形式
- **センサ群とのアクセス方法**：モバイル機器で利用するセンサ郡からのデータ取得方法
- **データ処理の統合方法**：データを取得する・サーバと通信する・画面出力するアプリケーションにデータを渡す・など一連のデータの流れを記述する方法

例えば，セカイカメラのようなモバイル機器の現在位置情報とカメラの向いている方角に合わせて，モバイル機器の画面上に情報を表示するアプリケーションを実装することを想定する．センサ情報を通信する際のデータ形式として，位置情報・方位情報の形式を定める必要がある．また，それらの情報を取得するために，GPS・加速度センサ・地磁気センサからデータを取得する必要がある．更に，Webサーバから現在位置に近い情報の取得・カメラ映像上への情報マッピング方法などの処理を記述する必要がある．

セカイカメラ以外のアプリケーションに対しても，これらの3つの要素について同様に適用することによって，提案モデルに従い容易に開発することができる．

これら3つの要素は，既存のフレームワークで対応することができない．本研究で提案するフレームワークでは，リアル指向型Webサービスの開発において独自に実装しなければならない3つの要素についてモデル化することを目指す．

3.2 提案フレームワークのモデル構成

本研究では，3.1節に示す3つの要素をモデル化するために，サーバ側とクライアント側の双方にMVCアーキテクチャを持たせたフレームワークを提案する．更に，双方のMVCアーキテクチャにおけるModelにセンサ情報の形式について共通の規定を設けることで，通信時のデータ形式を定義する．本研究で提案するフレームワークのモデル構成を図3に示す．図中のSensor Unitはモバイル機器に搭載されているセンサ群を想定している．

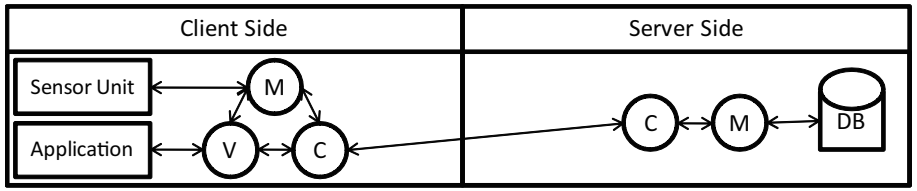

図 3 提案フレームワークのモデル構成

提案するフレームワークと既存のフレームワークとの相違点は，クライアント側のセンサ情報の取得と，通信する際のデータ形式をフレームワークに組み込んでいる点である．そのため，リアル指向型Webサービスの開発において，クライアント側の様々なセンサ情報を取り扱う処理をフレームワーク上で記述することが可能である．3.1節に示す3つの要素について新たにアプリケーションの構成を設計する必要がないため，開発コストの削減が見込める．

本フレームワークにおける要素のそれぞれの役割について，以下に記述する．

【クライアント側】

- **Model**：通信時のセンサ情報の形式を規定・利用できるセンサ群とのアクセスを抽象化
- **View**：ディスプレイへの表示・イベントをControllerに送信
- **Controller**：サーバとの通信・Modelからセンサ情報の取得・Viewへのデータ受け渡し

【サーバ側】

- **Model**：Webサーバ側のDBとの接続・センサ情報の形式の規定
- **Controller**：クライアントとの通信・Modelからデータの取得

提案するフレームワークにおいて最も特徴的なのは，クライアント側に存在するModelである．このModelはセンサ群とのアクセスを抽象化している．開発者はセンサ群に対して直接アクセスするのではなく，Modelを通じてアクセスし，データを取得する．更に，センサ情報について通信の際のデータ形式をModelで規定することで，センサに関する処理

ⓒ2009 Information Processing Society of Japan

を全て Model にまとめることができる．これにより，リアル指向型 Web サービスの開発において，既存のフレームワークで対応できなかった要素について対応できる．

4. フレームワークの設計

本章では，3.2 章で述べた提案フレームワークについて，設計の一例を述べる．提案するフレームワークの内部構造を図 4 に示す．

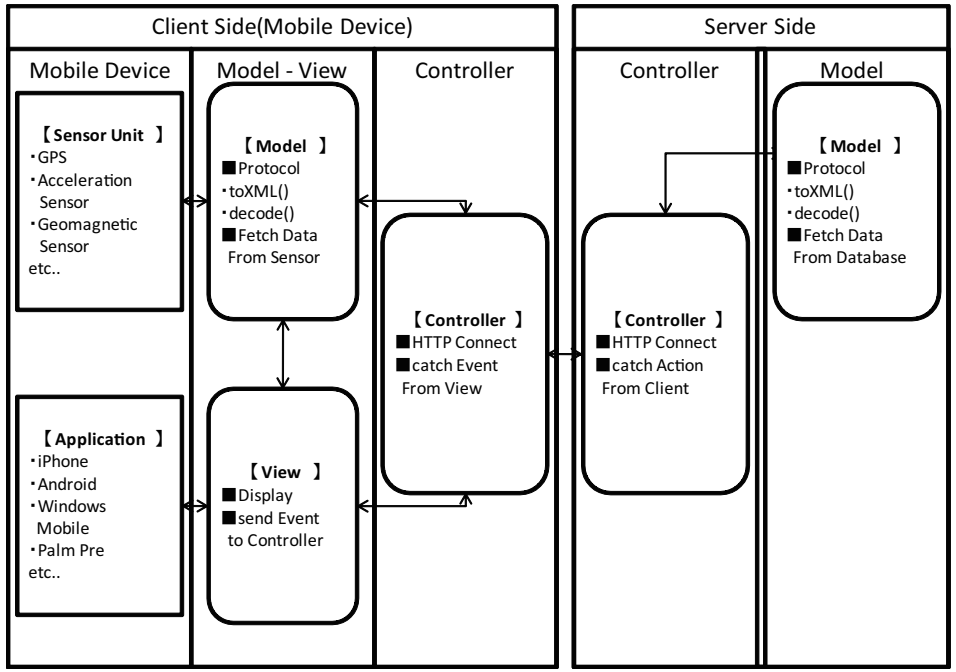

図 4 提案フレームワークの内部構造

図 4 では，クライアント側は Mobile Device, Model-View, Controller，サーバ側は Controller, Model の 5 つのブロックにそれぞれ分解できる．Mobile Device はモバイル機器のプログラムであり，センサ群とのアクセスや画面への出力を行うことができる．Model-View は提案するフレームワーク内において Mobile Device の要素とのやり取りを抽象化している．開発者は Model-View に対してプログラミングすることで，Mobile Device に対して直接プログラミングすることなく開発することができる．クライアント側とサーバ側の Controller は，それぞれデータの流れに関する処理を記述する．サーバ側の Model はデータベースとの接続についての処理を記述する．

前節で述べた Model, View, Controller については，以下のように実装する．

■クライアント側

【Model】

- **Protocol** : 規定されたセンサ情報の形式に従って，センサ情報と XML を相互に変換する機能である．例えば，位置情報について管理するオブジェクトに対して toXML リクエストを送ると，規定された位置情報の XML データが返される．更に，toXML リクエストを保持するオブジェクトは decode リクエストも受け取ることができ，これは XML のセンサ情報を元の情報に変換する機能である．Protocol 機能はサーバ側の Model も共通に持っており，センサ情報の取り扱いについて整合性を保つ．
- **Fetch Data From Sensor** : センサ群とアクセスしてデータを取得する機能である．モバイル機器で利用できるセンサ群とのアクセスを抽象化したオブジェクトを持つ．例えば，GPS と通信する処理をフレームワーク上で抽象化したオブジェクトに対して緯度経度取得などのリクエストを送ることでモバイル機器の現在位置情報を取得することができる．開発者はセンサに直接アクセスするのではなく，抽象化されたオブジェクトに対してリクエストを送ることでセンサからのデータを取得する．

【Controller】

- **HTTP Connect** : サーバ側の Controller と HTTP で通信を行う．センサ情報を XML 化して送受信を行うことができる．センサ情報以外の情報についても通信が可能である．
- **Catch Event** : View からリクエストを受け取る．クライアント側の Controller は View からリクエストを受け取ることが動作する．例えば View 側のボタンに Controller で定義したイベントを指定すると，ボタンにリクエストがあった時，指定したイベントが実行される．

【View】

- **Display** : ディスプレイへの表示に関する処理を記述する．例えば，ディスプレイ上にボタンを表示するために，ボタン配置リクエストを用いる．
- **Send Event** : ディスプレイ上で起こったイベントを受け取り，Controller に送信す

ⓒ2009 Information Processing Society of Japan

る．イベントとは，例えばボタンがクリックされた時など，特定の動作があった時に生じるリクエストである．特定のイベントが生じた時，クライアント側の Controller で呼び出すアクションを指定することができる．例えば，画面上に表示されたボタンを押すと，指定したイベントが呼び出される．

■サーバ側

【Model】

- **Protocol** : クライアント側の Protocol と同様である．
- **Fetch Data From Database** : データベースとアクセスしてデータの取得・更新をする機能である．データベースとの通信を抽象化したオブジェクトを持つ．これは，Ruby on Rails など既存の Web アプリケーションフレームワークにおける Model と同様の概念である．例えば，サーバ側で扱うことのできるデータベース上に，テキスト・画像・音声などに位置情報を関連付けた情報を保持するテーブルが存在するとする．この時，フレームワーク上ではテーブルに対応して問い合わせを行うオブジェクトを持つ．そのようなオブジェクトに対してリクエストを送ると，対応するテーブルに対してデータを照会する SQL 文を発行する．

【Controller】

- **HTTP Connect** : クライアント側の HTTP Connect と同様である．
- **Catch Action** : クライアント側の Controller からリクエストを受け取る．クライアント側の Controller からあるアクションを指定してリクエストを送ると，サーバ側の Controller に定義されたアクションを呼び出す．パラメータは HTTP の POST や GET で取得することができる．センサ情報の XML データもパラメータとして受け取る．

5. 考　　察

本研究で提案したフレームワークを用いることで，様々なリアル指向型 Web サービスが柔軟かつ低コストで開発できるようになる．例えば，サンプルとして 3.1 節で述べたセカイカメラのような機能を持つアプリケーションの開発について考察する．設計が必要な要件として，モバイル機器の位置情報・方向情報の形式の規定，GPS・加速度センサ・地磁気センサからの情報取得，Web サーバから現在位置に近い情報の取得・カメラ映像上への情報マッピングが挙げられる．この内，モバイル機器の位置情報の取得と現在位置に近い情報の取得の Controller における実装を Ruby on Rails の擬似コードで示すと，サーバ側は図 5，

```
class MyServerController
  def nearbyInfoAction
    # "location"パラメータから XML 形式の位置情報を取得
      locationParam = params[:location]
    # 位置情報を Location クラスのインスタンスに変換
      location = LocationModel.decode(locationParam)
    # 緯度経度を取得し，その付近にあるデータを取得
      nearbyInfo = InformationModel.findNearbyInfo(location.getLatLng)
    # 送信用のメッセージを XML の形式で生成して送信
      render LocationModel.toXML(nearbyInfo)
  end
end
```

図 5　ユースケース：サーバ側の実装

```
class MyClientController
  def buttonEvent
    # Location モデルから位置情報を取得
      myLocation = LocationModel.getLocation
    # 位置情報を XML の形式にして送信用メッセージを生成
      message = LocationModel.toXML(myLocation)
    # MyServerController の nearbyInfo アクションにメッセージ送信
      response = http.post('my-server/nearby-info', {"location" => message})
    # XML のレスポンスを Location クラスに変換
      nearbyInfo = LocationModel.decode(response)
  end  end
```

図 6　ユースケース：クライアント側の実装

クライアント側は図 6 のようになる．ここでは View と Model についての記述は省略する．

サーバ側のコントローラは，クライアント側のコントローラからリクエストを受けることで処理を開始する．サンプルの場合，サーバ側の MyServerController の nearbyInfoAction

ⓒ2009 Information Processing Society of Japan

に対し，クライアントが位置情報をパラメータとしてリクエストを送ることで，パラメータの位置情報の付近にある情報を取得する．擬似コード内には，位置情報を管理する Model として LocationModel クラスと，引数として与えられた位置情報の付近にあるデータを返すメソッド findNearbyInfo を持つ InformationModel クラスが存在する．LocationModel は，位置情報を表す Location クラスのインスタンスと，データ通信のための XML を相互に変換することが可能である．そのため，データ通信の際は XML に変換し，パラメータ取得後は Location クラスのインスタンスに戻すことができる．位置情報の XML 変換は Model 側で規定されており，開発者は通信時のデータ形式を気にすることなく通信部分のコードを記述することができる．

クライアント側にも LocationModel が存在し，サーバ側と違い XML に変換する toXML メソッドに加え，GPS にアクセスして位置情報を取得するメソッド getLocation を持つ．getLocation から得られた位置情報は Location クラスのインスタンスである．LocationModel を通して位置情報を取得するため，開発者はセンサとの接続を考慮する必要がない．

本研究で提案するフレームワークを用いて開発を行うことで，モバイル機器の位置情報の取得と現在位置に近い情報の取得がクライアントとサーバそれぞれ 4 行程度で実装することができる．このように，提案するフレームワークを用いることで開発コストを大幅に削減することができる．

6. ま と め

センサを搭載したモバイル機器の普及に伴い，センサ情報を活用するリアル指向型 Web サービスの開発は今後ますます重要になると考えられる．そうした背景を受け，本研究では，既存のフレームワークの問題点を踏まえてリアル指向型 Web サービスの開発を容易にするフレームワークを提案した．センサ情報の取り扱い方のモデル化のために，サーバ側とクライアント側の双方に MVC アーキテクチャを持たせたフレームワークを設計した．更に，センサ情報の通信のためのデータ形式をフレームワーク内で規定する仕組みを設けた．これにより，リアル指向型 Web サービスの開発者はセンサへのアクセスや通信の際のデータ形式を意識することなく開発することが可能になる．本研究で提案するフレームワークを用いることにより，リアル指向型 Web サービスの開発コストを削減することができる．

今後の課題として，設計したフレームワークの実装を行い，実行時のオーバーヘッドの評価や，実アプリケーションへの適用などが挙げられる．

参 考 文 献

1) Tonchidot Corporation: Air Tagging Device “Sekai Camera” (2008). http://tonchidot.com/Sekai_Camera.html
2) F. Buschmann, R. Meunier, H. Rohnert, P. Som-merlad, and M. Stal, Pattern-Oriented SoftwareArchitecture A System of Patterns, John Wiley & Sons, Ltd (1996).
3) Ruby on Rails http://rubyonrails.org/
4) 秋田 一郎, 満田 成紀, 福安 直樹, 吉田 敦, 鯵坂 恒夫：リッチクライアントに適した Web アプリケーションフレームワークの提案と実装，電子情報通信学会技術研究報告, Vol.104, No.570, SS2004-44, pp.7-12 (2005).

ⓒ2009 Information Processing Society of Japan