

IoT センシングサービスプロトタイピングフレームワーク

渡邊恵太^{†1} 赤塚大典^{†1} 関口直紀^{†2} 山口修平^{†2} 宇都宮栄二^{†3}

概要：IoT サービスの研究開発やサービスのための開発フレームワークが多数試みられているが、ハードウェアやネットワーク構成が関係するため、Web 開発のように手軽に参入できるほどのフレームワークがない。本研究では、ハード構成はできるだけ変えずに Web 標準技術やその開発環境を活用しながら、IoT サービスの開発のプロセスを整理し目的の問題解決のための設計や見積もりを低コストで行うフレームワークを提案する。

1. はじめに

環境をセンシングしそれらデータをインターネット上にアップロードし解析することによる、通知や予測といったサービスやアプリケーションを開発する IoT サービスの研究開発やサービスのための開発フレームワークが多数試みられている[1]。一方で、こうした IoT サービス設計や開発は、ハードウェアの構成とソフトウェアの構成、さらにネットワーク構成といった要素が絡み複雑である。さまざまなフレームワークが提案され、比較検証もされているが、Web 開発のように誰もが手軽に参入できるほどのフレームワークは未確立である。

こうした不確実な要素が多い環境においてはプロトタイピングが重要となる。プロトタイピングでは、素早いイテレーションによる試行性が重要となるため、ハードの要素検討を最小化しつつソフトウェア上で試行することが望ましい。そこで本研究では、スマートフォンをマルチセンサーデバイスとして捉え、これらを環境や測定したい対象に配置し、Web 開発のフレームやこれまでの効率的な開発モデルである MVC モデルを採用した IoT センシングサービスプロトタイピングフレームワークを提案する。

2. IoT センシングサービスプロトタイピングフレームワーク

IoT センシングサービスプロトタイピングフレームワークとは IoT センサーとしてスマートフォンを利用し、対象としたい環境内にそれを複数配置し、センシングデータを可視化し、MVC モデルでサービス開発を行う手法である。本手法によって、適切なセンサー配置さえ行えば、個別のセンサーモジュールを意識することなく、ソフトウェア上でセンサー値のイベント化や、意味の定義などのモデル化を行い、サービスの素早い構築が可能になる。ただし、前提として適切なセンサー配置が課題になる。本稿では、適切なセンサー配置を促すためのツール「配置比較ツール」を

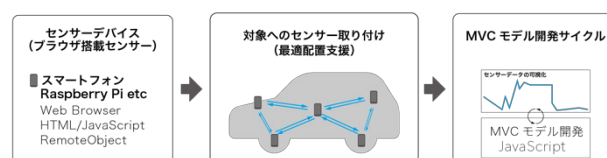


図 1 IoT センシングサービス
プロトタイピングフレームワークのプロセス

実装したので紹介する。

2.1 配置比較ツール

センサーデータはセンサーの配置位置によって変化する。そのため目的のセンサーデータを得るためには、配置の検討が必要になる。一方でセンサーの配置場所は、センシング対象や空間に見識のある専門家でなければ、配置場所選定が困難な場合がある。また、専門家であっても、探索的に配置決めるを行う場合がある。そこで配置比較ツールは、ある場所にセンサーデバイスを配置した際の、データを比較し、センサーがデータを取得できているか確認可能にする。確認は、各センサーデバイスの変化量を使用し比較する。最低 1 台からできる。1 台の場合は、ある A 地点に設置した場合と B 地点に設置した場合を置き換えることで、比較する。複数台ある場合には、センサー端末ごとに順位付けを行う。これにより、サービス設計者は変化量の少なすぎる場所や、センサー値がまったくはいつてこないセンサーデバイスの配置を別の場所への移動を検討することを支援する。

2.2 センサーデータとその取得

現在は、スマートフォンを使用している。またスマートフォンに入ったセンサーデータは Web ブラウザを通じて取得するため、Web ブラウザがサポートしていないセンサーデータは取得できない。また端末によっても搭載していないセンサーがある。現在取得できるデータは、たとえば加速度、磁気（方位）、近接センサ、音量（マイク）などがある。こ

^{†1} 明治大学

^{†2} 株式会社 KDDI 総合研究所

^{†3} KDDI 株式会社

これらのデータをすべて取得し、各端末はブラウザからサーバー上へデータを送信する。

2.3 実装

実装は Web ブラウザと JavaScript によって行った。スマートフォンのみならず、たとえば Raspberry Pi などのコンピュータボードでブラウザが動作すれば、センサーデータを取得できる。また、リモートオブジェクトとして実装した。リモートオブジェクトとして実装することで、遠隔地からそのデバイス上のプログラムを改変できるため、IoT のようなリモートでのセンシングが発生する環境におけるプロトタイピングにとって出向かず改変が可能になるため、合理的である。

2.4 データの可視化と MVC モデル

配置したセンサーからデータを取得したものを可視化する。設計者は、可視化したデータに基づき MVC モデルにおけるコントローラー部分でさまざまな処理を検討しながら、サービスを構築していく。より意味のある挙動の関数が定義できそうであればそれはモデルとして構築していく。たとえば、ドアの開閉をサービスとして使用したい場合には、以下のように記述する

■ デバイスの振る舞い（コントローラ）

```
class DoorSensor{
  isOpen(){
    if(compassmaxValue > 2000){
      return true
    }else{
      return false
    }
  }
}
```

■ アプリケーションロジック（モデル）

```
const thing = new Thing(DoorSensor, config);
if(thing.isOpen()){
  document.querySelector("#door").innerHTML = "open"
}else{
  document.querySelector("#door").innerHTML = "close"
}
```

■ 画面表示、デザイン（ビュー）

```
<style>
#door{
  font-size: 2em;
}
</style>
<div id="door">
</div>
```

3. 考察・議論

3.1 配置探索の試用

机の上に振動源を配置し、センサーデバイスを配置した。

各センサーデバイスは振動源に近い順に順位を表示した。これにより設計者は遠いデバイスや場所によるデータの特性による配置判断の探索を支援できる。

現在、配置比較ツールは変化量を軸として比較するが、ある時間の総量や最大値や最小値などを軸として評価する方法もあるだろう。

また、場所の探索には、設置したい対象における設置可能な箇所すべての場所が対象になる。すなわちその表面積に対してデバイス1つが占める面積を割る数が、探索可能場所になる。手持ちのセンサーデバイスの数が多いほど探索の時間は短くなるが、センサーデバイスを準備するコストが高まり、探索時間と金銭コストのトレードオフの関係がある。探索方法はこれから検討していく必要があるが、すべての場所を探索比較するには探索所要時間が高まることになり、一方で探索しない場所や、探索したがその時点では配置不要と判断したことが、あとになって問題にならないかといった問題がある。

他に、データ種類の特徴によっては配置の考え方も変わってくる。たとえば、ドアを開けたことを検知する仕組みをつくるといった場合は「ドアの開閉」という挙動に対するデータであることを教師ありデータとして設計者は理解できる。一方で、何かが起こるが何がおきるかは起きてみないとわからないような場合がある。

3.2 IoT サービスの見積もりの簡略化

IoT サービスを構築する場合、ある目的が定まっていたとしても、それが可能であるかどうかはハードウェアの選定や、ネットワーク環境、ソフトウェア構築の見積もりが必要になる。本研究では、目的に応じてハードウェアを構成しない戦略をとる。複数のセンサーセットは何をセットにするかは議論であるが、複数のセンサーを搭載することで、どれかが反応することを前提として、配置箇所の問題とソフトウェア上での試行錯誤によって、センサー値の意味を取得できると考える。この方法によって、複雑になりがちな IoT サービスの見積もりを設置対象の面積からどの程度の数のセンサーデバイスを配置するかといった見積もりを立てることができるようになるだろう。

4. おわりに

本研究では IoT サービスの設計や見積もりを低コストでの実現を目指した IoT センシングサービスプロトタイピングフレームワークについて紹介した。今後は長期利用を通じて課題点を整理し、実用的に利用可能なものを目指す。

参考文献

- [1] Developing IoT applications: challenges and frameworks, Udoh,Itorobong S. et al.IET Cyber-Physical Systems: Theory & Applications(2018), 3 65 <http://dx.doi.org/10.1049/iet-cps.2017.0068>