

卒業プロジェクト 2014 年度（平成 26 年度）

汎用的な Android Widget の作成及び それを用いた情報獲得手法の提案

慶應義塾大学 環境情報学部

中山 拓哉

増井俊之研究会

2015 年 1 月

卒業プロジェクト 2014 年度（平成 26 年度）

汎用的な Android Widget の作成及び それを用いた情報獲得手法の提案

論文要旨

天気予報や電車の運行情報などを Android Widget を用いて取得することは広く用いられている手法だが、Android Widget を用いてユーザーが望む情報がすべて手に入るとは限らない。しかも、既存の Android Widget を使用すると画面上に表示できる情報の数が限られてしまう。

そこで本研究ではユーザーが必要とする複数の情報を表示することの出来る汎用的な Android Widget を作成し、ユーザーが効率良く価値の高い情報を得ることが出来るようになることを目指した。

キーワード

Linda、Android、Android Widget

慶應義塾大学 環境情報学部

中山 拓哉

目次

第1章 序論	1
1.1 背景	1
1.2 目的	1
1.3 本論文の構成	1
第2章 研究内容	3
2.1 システム概要	3
2.1.1 データの取得と記録	3
2.1.2 ユーザーが受け取るデータの選択	4
2.1.3 ユーザーへの情報の提供	4
第3章 実装	6
3.1 サーバーサイド	6
3.2 Android クライアントサイド	6
3.2.1 データの取得	6
3.2.2 データの表示	6
第4章 考察	7
4.1 本研究についての考察	7
第5章 結論	9
謝辞	10
参考文献	11

図 目 次

1.1	現状の Widget の使用した画面	2
2.1	Android クライアントに送信するデータの選択画面	4
2.2	サーバーから送られる JSON の例	5
2.3	Android Widget として表示する例	5
4.1	葉山の風速情報	8

表 目 次

2.1 システム構成	3
----------------------	---

第1章 序論

1.1 背景

近年、スマートフォンなどのモバイルインターネット端末は爆発的に普及し、利用者の年代を問わず用いられるようになった。もはや生活の一部とも言える存在となったスマートフォンだが、私が日常的に使用する中で不便だと感じる点がある。それは Android Widget を用いた情報の獲得についてである。Android Widget は設置してさえおけばユーザーが能動的に操作をしなくとも、最新の情報を常に画面に表示してくれるので情報の獲得手法としては有効である。

しかし、Android Widget はいくつかの問題を抱えている。まず Android 端末の一つの画面に置くことの出来る Android Widget の数は限られているため、獲得できる情報の種類も限られてしまうという問題である。(図 1.1) 例えば天気予報を見なければ天気予報の Android Widget を画面に設置し、別の情報が必要であればまた別の Android Widget を設置する必要がある。Android Widget の性質上、取得したい情報に付随して不要な情報までもが表示され、それによって画面スペースが埋められてしまうというケースも存在する。また、自分の取得したい情報に対応した Android Widget が存在しない場合、自ら Android Widget を作成する以外の対応策がない。現状では Android Widget の作成は多数のスマートフォンユーザーにとって技術的なハードルが高く、問題の解決法としては現実的ではない。

本研究では単体で複数の情報を表示することの出来る汎用的な Android Widget を作成することでこれらの問題の解決を図った。

1.2 目的

本研究の目的は、ユーザーが必要とする複数の情報を表示するシステム及びインターフェースの実現である。それによって情報の獲得が容易になり、さらに取得する情報からノイズが減ることになり、ユーザーは効率良く価値の高い情報を得ることが出来るようになる。

1.3 本論文の構成

第1章では本研究の概要を書いた。

第2章では研究内容を説明する。第3章ではプロトタイプの実装方法を解説する。第4章では考察を書く。最後に第5章にて結論を書き本論文をしめることとする。添付として参考文献を追記する。



図 1.1: 現状の Widget の使用した画面

第2章 研究内容

本章では、本研究の内容を説明する。

2.1 システム概要

本システムの目的はユーザーが必要とする複数の情報を提供することで、ユーザーが効率良く価値の高い情報を得られるようにすることだ。サーバーは各種 API などを用いてユーザーが必要とする情報を取得し Linda¹へと書き込み、Android クライアントからのリクエストに応じて Linda からデータを取得し JSON 形式でレスポンスを返す。Android クライアントはユーザーの操作に応じてサーバーにリクエストを送り、サーバーから返された JSON をパースして表示する。

以下にシステムの構成を示す。

表 2.1: システム構成

システム	概要
Linda	データ元。データストリームからデータを取得する。
サーバー	Android クライアントからのリクエストがにに応じ Linda からデータを取得し Android クライアントに送信する。外部から取得したデータを Linda に書き込む。
WEB クライアント	Android クライアントに送信する情報を選択する為のビューを提供する。
Android クライアント	サーバーにデータを要求し、ユーザーに情報を表示する。
データベース	サーバーのデータを保存している。

2.1.1 データの取得と記録

データの取得元は三つ存在する。

まず一つ目として既存の API を用いて取得する方法である。天気予報や株価の情報など比較的ポピュラーな情報であれば既に API が存在するので、それらを用いて定期的にサーバーがデータを取得し Linda へ書き込む。

¹<http://linda.masuilab.org>

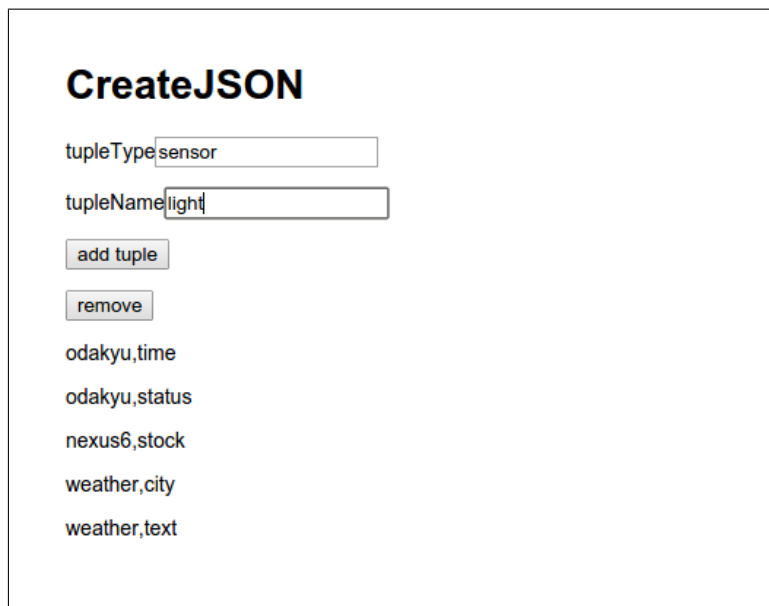
データをクラウド上で共有するためのフレームワーク。並列処理で同時に多くのクライアントを処理できる。

二つ目は API が存在しない Web ページ上の情報である。例として自分の利用する公共交通機関の運行情報や特定の商品の在庫情報などである。これらのデータは Kimono²を用いて取得したい部分に対応する API を作成しデータを取得する。

三つ目はセンシングデータである。自室などに設置した各種センサーからのデータに関してはセンシングする機器から直接 Linda に書き込むか、もしくはセンシングする機器からサーバー経由で Linda に書き込む。

2.1.2 ユーザーが受け取るデータの選択

ユーザーは受け取りたい情報を Web ページ上から指定する。Linda サーバーに存在するデータの中から TupleType と TupleName を指定する。(図 2.1)



CreateJSON

tupleType

tupleName

odakyu,time

odakyu,status

nexus6,stock

weather,city

weather,text

図 2.1: Android クライアントに送信するデータの選択画面

2.1.3 ユーザーへの情報の提供

ユーザーが選択した情報はサーバーによって取得され JSON 形式で Android 端末へと送られる。(図 2.2)Android は受け取った JSON をパースし Android Widget として表示する。(図 2.3)

サーバーへのリクエストは Android Widget の自動更新の際と、ユーザーが Android Widget をタップした際に実行される。

²<https://www.kimonolabs.com/>
指定した Web ページをスクレイピングし、API 化するサービス。

```
{
  "info": [
    {
      "0": "2015年01月16日14時43分現在",
      "1": "小田急線は平常どおり運転しております。",
      "2": "We are out of inventory. Please check back soon.",
      "3": "Fujisawa-shi",
      "4": "Mostly Cloudy"
    }
  ]
}
```

図 2.2: サーバーから送られる JSON の例



図 2.3: Android Widget として表示する例

第3章 実装

この章では本研究でのプロトタイプを実装し、その流れを解説する。

3.1 サーバーサイド

実装に際してサーバーサイドはNode.js¹を用いて実装した。またデータベースにはmongoDB²を採用している。

言語は Javascript を使い、Node.js の socket.io³で実装した。mongoDB にはユーザーが指定した TupleType と TupleName のペアを保存している。

Linda サーバーに対して socket.io を用いて常にコネクションを貼り、Android クライアントからのリクエストに応じて、mongoDB に保存されている TupleType と TupleName のペアを元にデータを取得している。

3.2 Android クライアントサイド

3.2.1 データの取得

言語は Java を使い、サーバーへのリクエストには JSON 形式のデータの取得を容易に行うために Volley⁴を使用した。

また Android Widget の自動更新は最短でも 15 分間隔のため、取得したいデータの種類によってはもっと新しい情報が欲しいという状況が想定される。そのため表示内容の手動更新機能をつけた。表示内容を手動更新する操作には Android Widget のタップを採用することで、情報を表示するスペースを圧迫することを回避した。

3.2.2 データの表示

データの表示に関してサーバーから受け取った JSON は一要素ずつ Android Widget 上の textview に挿入され表示される。

¹<https://nodejs.org/>

²<http://www.mongodb.org/>

³<http://socket.io/>

⁴Android の HTTP 通信用ライブラリ。ネットワーク通信処理やキャッシュ処理を簡単に行うことが出来る。

第4章 考察

4.1 本研究についての考察

今回作成したシステムを用いることで現状の Android Widget の問題をいくつか解決することができた。多くの Android Widget を置くことで画面スペースを占有してしまう問題に対しては、情報をテキストとして並べることで省スペースを実現した。必要な情報に付随して不要な情報も表示されるという問題があったが、ユーザー自身が表示される情報を指定するため、不要な情報は表示されなくなった。自分の取得したい情報に対応する Android Widget が存在しない場合、自ら Android Widget を作成するしか対応策が無かったが、Web ページ上のテキストデータやセンシングデータなどを取得可能になった。これらの面からユーザーに効率良く価値の高い情報を提供するという目的は達成できた。

しかし、今回作成したシステムを実際に利用した結果として以下のような問題点が発見された。まず、テキスト形式のデータしか扱う事ができないが、取得したい情報がテキスト形式のデータとは限らないという点である。(図 4.1) は葉山の風速情報であるが、テキスト形式のデータではなく風速計をライブカメラで撮影した映像によって配信している。今回作成したシステムではこういった画像や映像で配信される情報を取得することができない。

次に、取得したデータの表示方法に課題が残る。今回作成したようなプレーンなテキストだけを、ただ並べるだけのインターフェースが最善とは言えない。見やすさや分かりやすさを意識し、インターフェースを改善することが今後の課題として見つかった。



図 4.1: 葉山の風速情報

第5章 結論

本研究では、ユーザーが必要とする複数の情報を表示するシステム及びインターフェースの実現を試みた。結果として必要な情報に付随して不必要な情報が表示される問題や、多くの Android Widget を置くことで画面スペースを占有してしまう問題、自分の取得したい情報に対応する Android Widget が存在しない場合、自ら Android Widget を作成するしか対応策が無い、といった問題を解決することができた。

そして、第4章で述べたようにいくつかの課題が見つかった。さらに効率のよい情報獲得の目指し、これらの課題の解決に今後も取り組んでいきたい。

謝辞

本研究を進めるにあたり、数々のアドバイスを頂いた増井俊之教授に深く感謝申し上げます。
また、研究会のメンバーにも多くの支援を頂きました。ありがとうございます。

中山 拓哉

参考文献

- [1] Android Developers. <http://developer.android.com/>.
- [2] java. <https://www.java.com/ja/>.
- [3] Linda. <http://linda.masuilab.org/>.
- [4] mongoDB. <http://www.mongodb.org/>.
- [5] node.js. <http://www.nodejs.org/>.
- [6] socket.io. <http://socket.io/>.
- [7] 椎尾勇樹, 渡邊雄一, 松本真佑, 佐伯幸郎, 中村匡秀. ホームネットワークシステムにおけるプレゼンスセンシングのためのサービスフレームワークの提案. 電子情報通信学会技術研究報告. IE, 画像工学, No. 113(211), pp. 1–6, 2013.