

# Web of Things アーキテクチャの農業分野への試験的実装

渡邊 駿介<sup>†</sup> 顧 奕豊<sup>†</sup> 小野寺 陽平<sup>†</sup> 平尾 龍也<sup>‡</sup> 藤井 章博<sup>‡</sup>

法政大学理工学部<sup>†</sup> 法政大学院理工学研究科<sup>‡</sup>

## 1. はじめに

近年、日本の農業の現場では農業従事者の高齢化が急速に進み、労働力不足が深刻な問題となっている。問題解決のため、IoT(Internet of Things)の導入による農作業支援が期待されている。しかし、IoT では異なるシステム、異なる機種間で相互運用性が確保されづらい、システムの孤立(サイロ化)という問題が提起されている。W3C はサイロ化を避ける目的で、Web 技術を IoT の運用に活用する WoT(Web of Things)アーキテクチャ<sup>[1]</sup>を提唱している。本稿では、WoT アーキテクチャの試験的実装と、その評価に関する報告を行う。

## 2. 研究概要

以下の図 1 に W3C が提起している WoT システムの抽象アーキテクチャの構成図を示す。

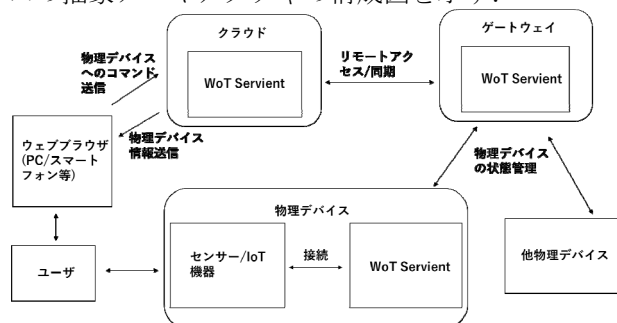


図 1.W3C の WoT の抽象アーキテクチャ構成図

図 1 で示されている WoT Servient とは、サーバとクライアントの役割を併せ持つ一種のゲートウェイである。この抽象アーキテクチャが WoT システムの一般形となる。このアーキテクチャに沿って我々は実装例として以下に述べる三種類のシステムを実装した。

一つ目は、農作物の生育環境のデータ収集を行うシステムである。センサーデバイスで測定したデータを、HTTP プロトコルを用いて Web サーバに送信し、データベースへ格納する。

二つ目のシステムは、一つ目と同様、生育環境のデータ収集を行う。こちらは、MQTT プロトコル<sup>[2]</sup>を用いて通信を行う。さらに、AWS(Amazon Web Service)を活用してデータ分析を行い、データの数値が閾値を超えた場合に警告メールを送信する機能を実装した。

三つ目のシステムは、人物の接近を検知し、メールでユーザに侵入を知らせるシステムである。センサーで人物の接近を検知し、カメラで画像を撮影する。サーバで人物像の識別が行われ、人物の接近が認められた場合、メールでユーザに通知される。

## 3. システム概要

### 3.1. HTTP プロトコルを用いた環境データ収集システム

本項のシステムはセンサーデバイス、Web サーバ、DB サーバで構築される。センサー値をデバイスから HTTP リクエストによって Web サーバへ送信し、受け取った Web サーバがデータベースに格納する。システムの構成図を以下の図 2 に示す。

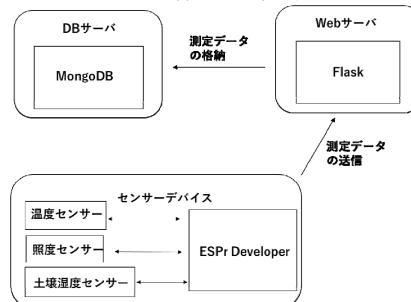


図 2. 環境データ収集システムのシステム構成図

センサーデバイスは、開発ボード ESP Developer<sup>[3]</sup>に温度センサー、照度センサー、土壌湿度センサーを接続して作成した。デバイスの機体名、測定日時、測定した環境値を JSON データにまとめ、Web サーバへ HTTP 通信で送信する。

ESP からデータを受け取るために Web サーバを構築した。開発言語は Python、フレームワークは Flask を使用した。また、データを格納するデータベースには、オープンソースソフトウェアのドキュメント指向データベースである MongoDB を採用した。それに関連して、サーバからデータベースを操作するために MongoDB クライアントの

Experimental implementation of the Web of Things architecture to the agricultural sector

<sup>†</sup>Shunsuke Watanabe <sup>†</sup>Ekihiko Ko <sup>†</sup>Youhei Onodera

<sup>‡</sup>Tatsuya Hirao <sup>†</sup>Akihiro Fujii

<sup>†</sup>Hosei University Faculty of Science and Engineering <sup>‡</sup>Hosei

University Graduate School Graduate Faculty of Science and Engineering

PyMongo を使用した。ESPr からデータを受信したサーバは JSON データをデータベースへ格納する。

### 3.2. MQTT プロトコルと AWS を用いた環境データ収集、データ分析システム

本項のシステムはデータ収集を行うセンシング部分とデータ分析を行うクラウド部分に分けられる。センシング部分は、測定を行うセンサーデバイスと測定データを受信しクラウドへ送信する Web サーバで構成される。クラウド部分は AWS を用いて設計した。システムの構成図を以下の図 3 に示す。

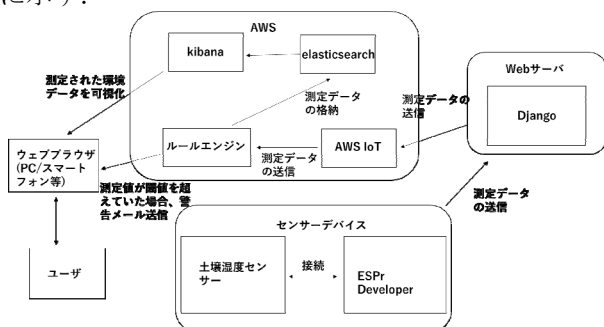


図 3. データ収集、分析システムのシステム構成図

図 2 で示したシステム同様にセンサーデバイスは ESPr Developer を用いて作成した。測定したデータを JSON データにまとめ、MQTT 通信で Web サーバへ送信する。

ESPr からデータを受け取るために Web サーバを構築した。開発言語は Python、フレームワークは Django を使用した。また、Python で MQTT 通信を行うためのライブラリ、Mosquitto を使用した。ESPr からデータを受け取った Web サーバは JSON データを MQTT 通信でクラウドに送信する。

クラウドに送信されたデータは AWS の Elasticsearch Service に格納され、データ解析ツール kibana で可視化することができる。また、データを AWS IoT のルールエンジンを利用することで分析し、設定した閾値を超えた場合に設定したメールアドレスに警告メールを送信出来るように実装した。図 2 に、土壌の水分量のデータが閾値 300 を下回った場合に送信される警告メールを示す。

### 3.3. ネットワークカメラを用いた人物接近検知システム

本項のシステムは、カメラデバイス、センサーデバイス、サーバで構築される。センサーで人の接近を検知した場合にネットワークカメラで撮影を行い、サーバで人物像の検出を行い人物の侵入検知を行う。システムの構成図を以下の図 4 に示す。

センサーデバイスは ESPr Developer に赤外線センサーを接続して作成した。センサーで人の接近を検知した場合、カメラデバイスに HTTP 通信で通

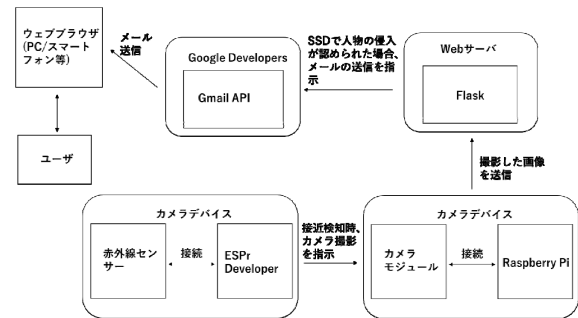


図 4. 人物接近検知システムのシステム構成図

知を行う。カメラデバイスは、シングルボードコンピュータ Raspberry Pi<sup>[4]</sup>に専用のカメラモジュールを接続して作成した。センサーデバイスから人の接近の通知を受信したカメラデバイスは撮影を行い、画像をサーバへ送信する。

データを受け取り、画像の識別を行う Web サーバを構築した。開発言語は Python、フレームワークは Flask を使用した。画像を受け取ったサーバは、SSD(Single Shot Multibox Detector)というアルゴリズムを用いて人物像の識別を行う。人物の侵入が認められた場合、警告メールによって侵入を通知する。メールの送信には、Google Developers の Gmail API を使用した。

## 4. まとめ

本研究で、試験的に WoT アーキテクチャを用いた農作業支援のためのシステムを三つ構築した。システムの構築において、通信のプロトコルやデータ形式などにおいて、既存の Web の技術を利用した開発を行ったため、使用する技術の選定や、開発者間での知識の共有をスムーズに進めることが出来た。三つのシステム全てで Web 技術を用いた通信を行っているため、連携を行うことが期待できる。今後の実装で相互運用性の検証を進め、サイロ化が解消されることを実証していく。

我々はこれまでシステムの試験的な実装を進めてきた。今後実際に農業の現場への導入を進めていく中でシステムの改善や機能の追加などを行っていく。また、農作業支援で開発されたシステムを他の分野で活かす、または他の分野で使用されているシステムと連携して運用していくような用途も併せて検討していきたい。

## 参考文献

- [1]WEB OF THINGS AT W3C, <https://www.w3.org/WoT/>, (2018年1月11日閲覧).
- [2] MQTT, <http://mqtt.org/>, (2018年1月11日閲覧)
- [3] ESPr® Developer (ESP-WROOM-02 開発ボード) - スイッチサイエンス, <https://www.switch-science.com/catalog/2500/>, (2018年1月11日閲覧).
- [4] Raspberry Pi - Teach, Learn, and Make with Raspberry Pi, <https://www.raspberrypi.org/>, (2018年1月11日閲覧).