小規模工場のスマート化に向けた取り組み
―マイクロサービスで構築する自動化システム―

千家　雅之（情報・生産技術部 システム技術グループ）

1. はじめに

近年、スマートファクトリは生産現場と経営層が一体となって事業変革を推し進める枠組みとして期待されており、既存工場のスマート化が大企業を中心に進められている[1]。一方、小規模工場のスマート化は積極的に取り組まれているとは言い難く、費用対効果への懸念やデジタル人材不足が障壁になっていると考えられる。システムの委託開発に伴うベンダーロックインがそれらの障壁を強固にしている可能性があり、委託開発と自社開発を容易に使い分けることができればスマート化に着手しにくい状況が改善されるとみられる。そのためには柔軟なシステム構築手法が求められる。その一つにマイクロサービスアーキテクチャがあり、本稿では、一例として外観検査システムを取り上げ、マイクロサービスで構築する自動化システムの可能性検証を行う。

2. マイクロサービスアーキテクチャ

ソフトウェア開発では、大規模で複雑であるほど独立して交換・更新が可能なソフトウェアの部品であるコンポーネントを用いた開発が重要視される。コンポーネント化によってソフトウェアの保守性や拡張性の向上が期待できる。しかし､コンポーネント化されたソフトウェアであっても、開発者に代わって不具合修正や機能拡張に対応するには多くの労力が必要であるため、ベンダーロックインに陥りやすい状況は改善されない。マイクロサービスアーキテクチャは、ソフトウェアからコンポーネントを単独で動作可能なマイクロサービスとして切り出し、そのマイクロサービスを連携させて一つのアプリケーションを実現する形態であり、クラウドベースのアプリケーションでは広く取り入れられている。マイクロサービスアーキテクチャの採用によってアプリケーションが内包する機能の境界が明確になるため、その境界であるインターフェースの規定に従った通信さえできれば、マイクロサービスの中核的なプログラムは自由に作成することができる。つまり、A 社が作ったマイクロサービスから B 社のものへ変更することが可能になるため、ベンダーロックインの回避が容易になる。

3. ステートレス通信と Web API

マイクロサービスアーキテクチャにおいてマイクロサービスを利用する側（クライアント）とマイクロサービスとの結合度を小さくすることが重要となる。そのためには、クライアントとサーバの通信において、サーバは各リクエストを個別に処理し、前のリクエストの状態を保持しないステートレス通信が求められる。マイクロサービスの API には様々なものが用意されているが、容易にステートレス通信を実現する API として Web API の一種である REST API があり、広く利用されている。

4. 簡易外観検査システムの構築

4.1 前提

外観検査システムとは､カメラや画像処理装置を用いて、製品や部品の表面を確認し、キズ、汚れ、変形、欠け等の外観上の欠陥を検出するためのシステムで、多くの工場で導入されている。本稿における簡易外観検査システムは機能検証用の最小限の構成とし、実装の対象は外観検査システムの中で最も基本的な機能（搬送装置による検査対象物の搬送、カメラによる撮影、画像解析による良否判定、不良品の分別、およびシステム全体の制御）のみとした。

4.2 システム構成

本システムは、備えるべき機能を機器別に分け、1 台のシングルボードコンピュータ（以降、SBC）上に下記のマイクロサービスとして搭載する構成とした（図 1）。

(a) ベルトコンベアによる検査対象物の搬送
(b) カメラによる撮影、画像解析による良否判定
(c) ロボットアームによる不良品の分別（押し出し）

また、システム全体の制御は Web アプリケーションで行う

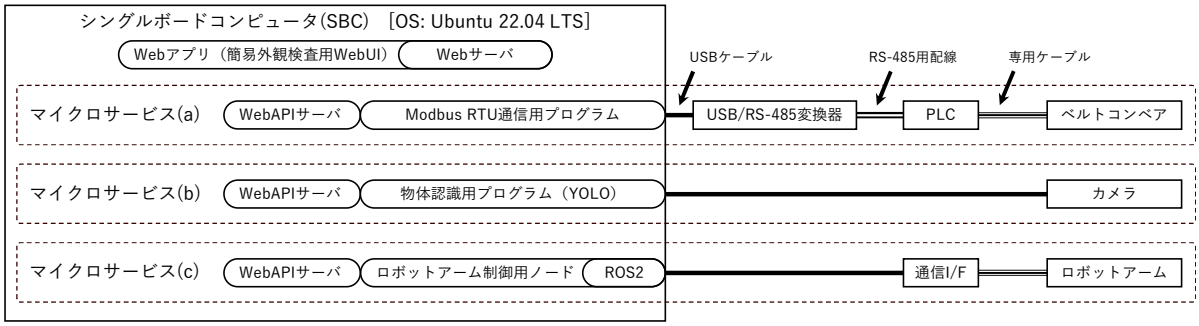

図 1　システム構成図

実装とした。Web API には REST API を採用した。

本システムを構成する機器は図 2 の通りで、ロボットアームに ROBOTIS 社製 OpenMANIPULATOR-X、カメラに DOBOT 社製 DoVision、ベルトコンベアにバイナス社製 BSU-1003 を用いた。また、SBC には NVIDIA 社製 Jetson AGX Orin Dev Kit、PLC には Arduino Opta、USB/RS-485 変換器にはラトックシステム社製 REX-USB70 を使用した。

4.3 各マイクロサービスの詳細

マイクロサービス(a)から(c)に統一的な手段でアクセスするために各マイクロサービスに図 3 に示す共通のステートマシンを組み込んだ。図 3 に記載の遷移条件は、例えば Initiated 状態から Ready 状態に遷移するには下記エンドポイントに GET メソッドでアクセスすることを意味する。

```
http://{WebAPI_BASE_URL}/initialize
```

InProgress 状態から Executed 状態への遷移は機器の動作が完了したら自動的に遷移する。また、各マイクロサービスの状態は、/get-state のエンドポイントパスから得られる。これらの共通の API は REST API を構築するためのフレームワークである FastAPI を用いて実装した。

次に各々のマイクロサービスで実行されるプログラムについて説明する。マイクロサービス(a)はベルトコンベアを直接的に制御する PLC 上の制御プログラムと、SBC と PLC 間の Modbus RTU 通信用プログラムにより構成され、それぞれ ST 言語と Python で実装した。マイクロサービス(b)は YOLOv8 を用いた物体検出用プログラムと 100 枚のアノテーション済み画像を用いて YOLO8n モデルからファインチューニングしたモデルで構成され、Python で実装した。マイクロサービス(c)はロボット用ミドルウェアである ROS2、その上で動作するロボットアーム制御用ノード、FastAPI と連携するための rosbridge により構成され、ROS2 に準じた方法で実装した。

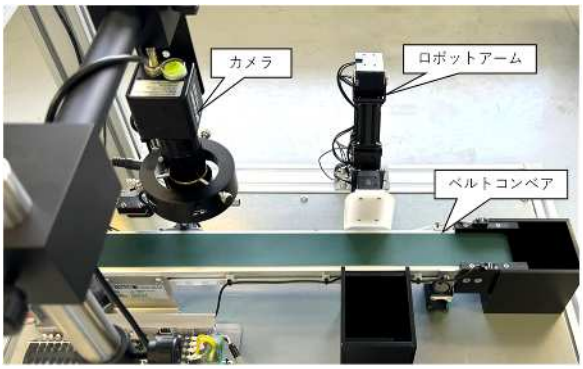

図 2　簡易外観検査システムを構成する機器

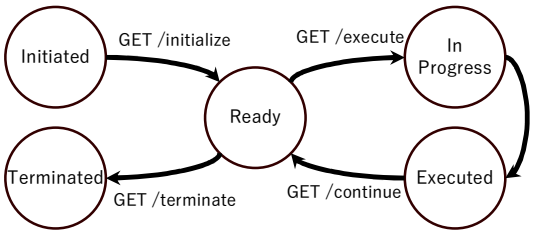

図 3　マイクロサービスのステートマシン図

4.4 Web アプリケーション

Web アプリケーションはフロントエンドのビルドツールである Vite と JavaScript ライブラリである React を用いて構築した。図 4 の操作画面では、各機器の INITIALIZE、CONTINUE、TERMINATE のボタンは図 3 の/initialize、/continue、/terminate に該当し、押下することで状態を遷移することができる。また、START ボタンを押すと外観検査の一連のプロセス（搬送開始、搬送停止、撮影・画像解析・不良品の分別、搬送開始、…）が実行される。図 5 は外見検査の実施例であり、ペットボトルキャップの汚れを検出して不良品判定をしている時の画面である。

5. おわりに

本稿ではベンダーロックインを避ける方法の一つとしてマイクロサービスアーキテクチャに着目し、そのアーキテクチャのもとで簡易外観検査システムを構築した。個々のマイクロサービスに機器の状態を表すステートマシンを内包することでステートレス通信でありながら外観検査の一連の動作を実現することができた。特定のベンダーに依存せずにソフトウェアや機器の変更が可能になる仕組みを確認した。この仕組みはマイクロサービス単位で作業を分担させることができるため、中小製造業における複数人での自社開発にも適用可能であると考えられる。

本稿での実装は 1 台の物理マシンに 3 個のマイクロサービスが動作する構成であったが、今後は、コンテナ仮想化技術を用いてマイクロサービスをコンテナ化して可搬性を高めて、その有用性や問題点を明らかにする予定である。

【参考文献】

(1)　経済産業省，厚生労働省，文部科学省：「2024 年版ものづくり白書」(2024-5)

図 4　Web アプリケーションの操作画面

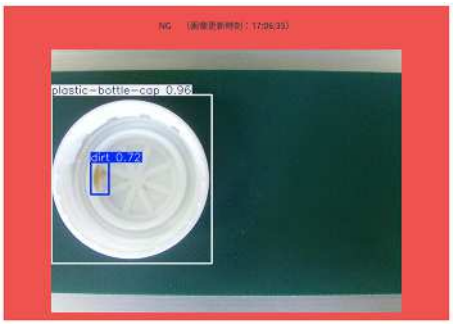

図 5　不良品判定時の画面