Usage of MIDIH opensource tools in production industry: *production order execution with robotic arm demonstrator*

Overview

In a traditional process-wise toolchain, path from planning to physical execution consists a list of manual operations that need to be completed by process handlers. In general, those operations cover defining, addressing, synchronizing, executing and supervising the processes. Effectiveness of this approach is limited by custom means of communication and human labor capacity. Modern factories are not free from usage of traditional paper forms, carrying the data to endpoint operators of production process or machines.

In contrast to the described approach, project implementation demonstrates direct interface between the service customer and the machine. Concluded architecture aims in MIDIH objectives through the digitalization of the product and cutting off the long chain od decision making process, using an interconnectable system based on Arrowhead Framework. Potentially within the MIDIH ecosystem, clients may crossover their services through factories – e.g. directly ordering type of material from available suppliers or selecting goods from various and competitive subcontractors.

In prepared demonstrator the leading scenario for implementation is a representation of a micro-production line (the robotic arm) that completes a production order in accordance to demand from a system client, who selects and “consumes” a predefined service, which acts as an interface with the processing line.

Implementation

The Arrowhead framework assumes that any technological process (in the context of the industry), any action carried out by an "agent" (e.g. motion of a machine-controlled machine), or any other executive system, can be abstracted as a service, i.e. we can assume that each such a system is a "provider" (a supplier) of some service . From this assumption we can conclude that all complex systems can be divided into "suppliers" of various services (e.g. each machine in a production line is a supplier of a service). The definition of a "consumer" of a service is also introduced - it could be a program, launched by an autonomous system or a human-controlled interface. Having established this foundation, it is possible to create a Service Registry, as well as a system that manages the interactions between "providers" and "consumers", which is precisely what Arrowhead Framework aims to achieve.

Usage

User perspective in system interface for the assumed production scenario:
- Client selects one of the available services from the user interface of the “consumer” program
(in this case – one of two preprogrammed moves),
- Consumer program sends an appropriate request to the Arrowhead Core (in the background),
- Arrowhead Core system finds a matching service provider (assigned to production workplace) and returns its address to the consumer program (in the background),
- Consumer sends client’s request directly to the provider (in the background),
- Provider executes the incoming request on the connected production hardware,
- After physical execution, client gets information about success or failure (this is achieved using mounted sensors).
- System waits for workplace items to be reloaded, to set the availability of service.

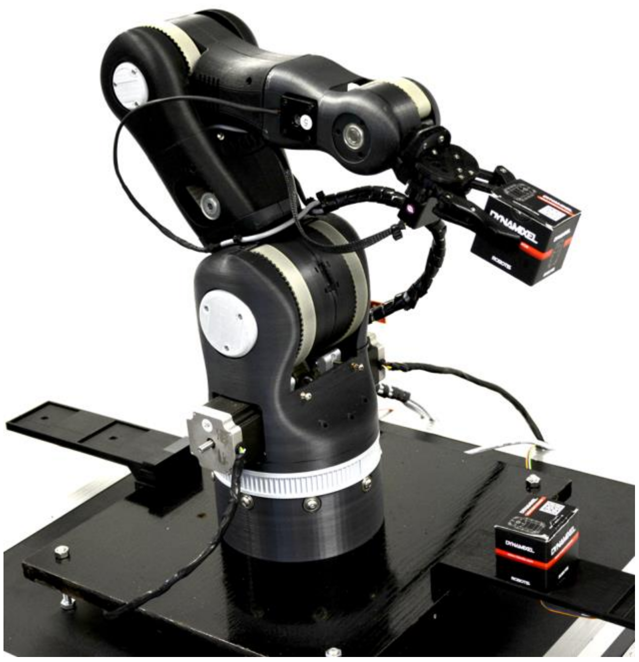

Figure 1 Pilot platform - robotic arm

Three main system benefits:

1. **Client does not care about place of production, selects only service or its kind (process automatization).**
2. **Client and production center maintain continuous status communication.**
3. **System is scalable through plant processes or between MIDIH nodes.**

Toolchain and robotic platform

Below is a list of steps taken to build the presented setup:
1. Parts of the robotic arm from an open source project were 3D-printed,
2. The parts were assembled, connected to stepper motors and other appropriate mechanical parts, the stepper motors were connected to DC motor drivers,
3. Two platforms (an “input” and an “output”) were 3D-printed and attached to the base on which the robotic arm was placed,
4. Four optical sensors were soldered to custom-etched PCBs, and then attached to the “input” and “output” platforms (two on each),
5. An Arduino board with custom-written firmware was used as a controller of the arm – optical sensors were used as inputs, and stepper motors as outputs,
6. A Raspberry Pi was used as a platform running customized code based on the “service provider” skeleton from Arrowhead Consortia’s GitHub,
7. Arrowhead Core Systems were run on an on-premises server running Ubuntu,
8. A custom “service consumer” providing a web interface was written and run on another on-premises server running Ubuntu,
9. Another computer was used as a “client machine”, accessing the web GUI of the “service consumer”.

Possible improvements

1. Adding more functionalities (more preprogrammed trajectories as service) to the “production line” and modifying the GUI accordingly.
2. Adding more “production lines” (more devices providing the same service or its kind.
3. Adding support for secure connections (e.g. using SSL).
4. Adding support for additional Arrowhead Systems (e.g. EventHandler for handling asynchronous events)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 767498.

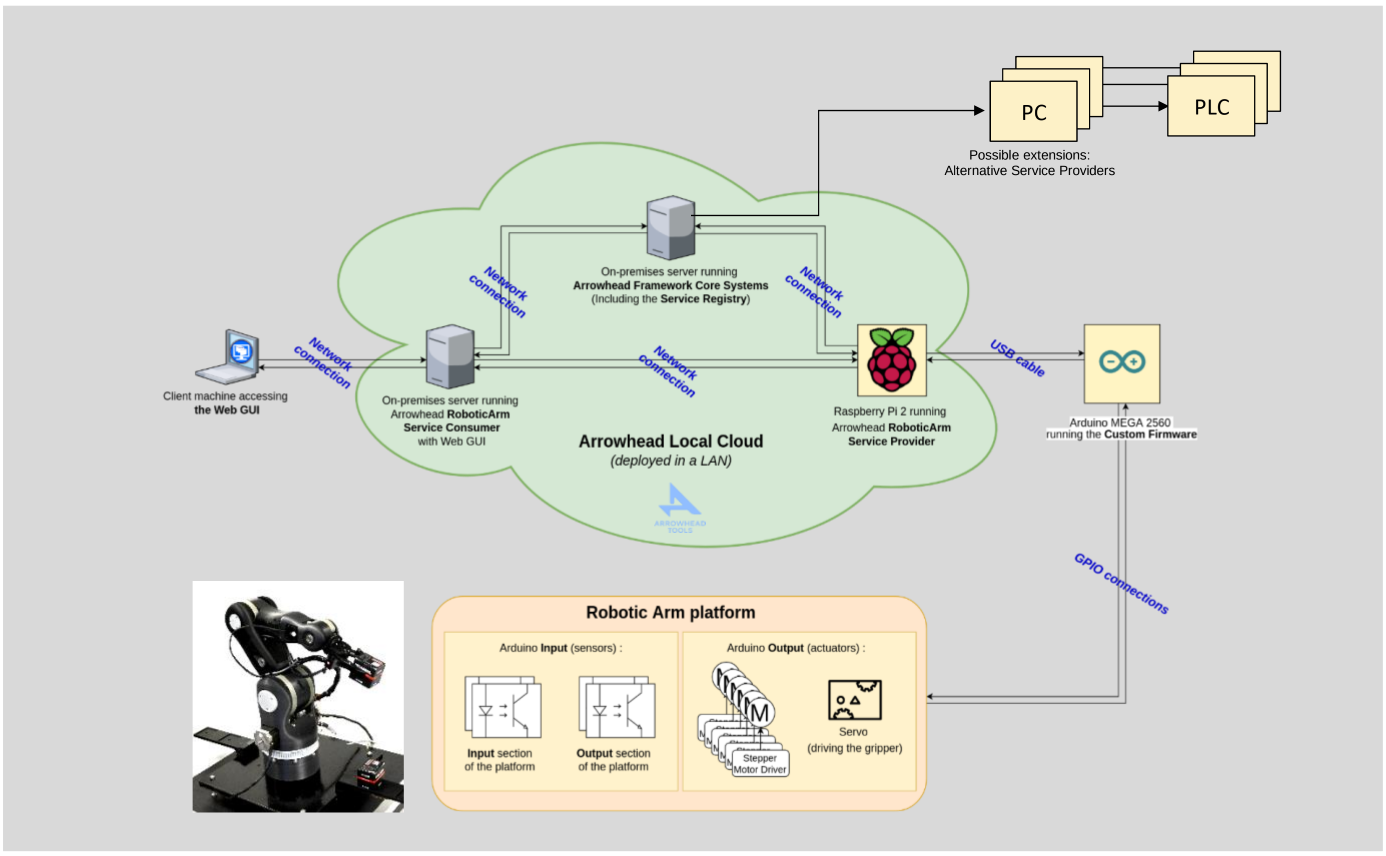

Figure 2 Architecture Diagram