Dynamic and Flexible Data Acquisition and Data Analytics System Software Architecture

David Hästbacka
Tampere University
Tampere, Finland
david.hastbacka@tuni.fi

Jari Halme, Martin Larrañaga
VTT Technical Research Centre of Finland
Espoo, Finland
{jari.halme|martin.larranagaunanue}@vtt.fi

Rupesh More, Heikki Mesiä, Mikael Björkbom
Konecranes Global Corporation
Hyvinkää, Finland
{rupesh.more|heikki.mesia|mikael.bjorkbom}
@konecranes.com

Laurentiu Barna
Wapice Oy
Vaasa, Finland
laurentiu.barna@wapice.com

Henri Pettinen, Marko Elo
CrossControl
Tampere, Finland
{henri.pettinen|marko.elo}@crosscontrol.com

Antti Jaatinen, Henrikki Hoikka
Metso Corporation
Tampere, Finland
{antti.jaatinen|henrikki.hoikka}@metso.com

***Abstract*—The introduction of new sensors and affordable measurement devices increases the amount of data used in cyber-physical systems and various sensing applications. These systems often grow extensive encompassing not only different kinds of data sources and sensors but also different kinds of use cases for the data. This paper presents a sensor system application architecture outlining the requirements on software and interfaces for connecting different kinds of measurement devices, executing computational functions on various hardware platforms, and connecting this information to various use case applications. Key objectives for the concept include dynamic orchestration of information flows and flexible configuration where computational functions are executed, support for different data sources and hardware, and utilization of computational resources on all levels from edge to cloud. The system application architecture designed is considered for two application domains within the Industry 4.0 paradigm.**

***Index Terms*—measurement system, data analytics, cyber-physical systems, dynamic orchestration, system architecture, system requirements, edge computing, arrowhead framework**

I. INTRODUCTION

Process data is inevitable for efficient production as well as maintenance and management of industrial equipment and assets. For years process data has also been used for purposes such as predictive maintenance or creation of value added services such as performance optimisation or adjustment to operational context. For complex cyber-physical systems (CPS) the value adding services offer true benefits to the operator or owner of the equipment - especially when manufacturer data from a broader fleet can be utilised in the analytics.

Complementary sensors and data sources are often needed to fulfill these needs. This means that there is not only one input for sensor data and the information can stem from other auxiliary systems as well. The challenge lies in integration of these sources of data in an efficient and interoperable way with regard to data format and semantics, protocols as well as mechanisms for connecting the different system components.

In addition, security is emphasized when systems and applications from different vendors and companies need to be integrated, often within the same networks. The challenge is even greater when this information is brought to use in Industrial Internet applications such as those envisioned as part of Industry4.0 in which components can be easily exchanged in a plug and play manner and systems collaborate flexibly. The integration challenge spans further to information security which in the case of dynamic interactions calls for open and agreed practices to ensure the integrity and security required.

Modern sensor and data analytics application systems aimed for such environments also need to construct their internal components similarly, i.e. dissect functionality into smaller independent services for preprocessing, storage, advanced analytics, and routing information between components distributed on edge, fog and cloud levels.

The main contributions of this paper are in 1) identifying the software and interface requirements for modern sensor and data analytics application systems and 2) outlining the software architecture and system integration mechanisms for flexible and dynamic configuration of information streams and computational services using the Arrowhead Framework [1].

II. RELATED WORK AND BACKGROUND

Architectures for maintenance services for CPS have been presented in [2]–[4], a Maintenance4.0 framework was described in [5] and a concept for integrating condition monitoring to maintenance operations was introduced in [6]. Internet based measurement and monitoring of production assets related to this paper has been previously discussed in [7].

The Arrowhead Framework has been developed as an infrastructure for services in various fields of application, initially for the IoT integration of local clouds [8]. A graph based system of systems composition model has already been proposed with identified requirements for industrial information distribution [9], while similar needs for other domains have been presented e.g. for energy usage monitoring [10] and utility networks monitoring [11].

III. REQUIREMENTS FOR SENSOR AND ANALYTICS SOFTWARE APPLICATION SYSTEM

The concept has been designed based on two use cases described in [7] and also detailed in the following subsections.

A. Use Cases in Production and Condition Monitoring

1) Production and assembly monitoring: This use case aims at creating an intelligence system that would provide end-to-end process visibility and task optimization recommendations for material handling production processes.

The solution uses a highly decoupled architecture based on SOA (Service Oriented Architecture) and edge computing concepts for gathering, processing, and analyzing data and for providing monitoring and intelligence services to factory floor staff. The implementation involves enhancing the material handling equipment with data gathering edge devices consisting of several sensors and measurement devices for monitoring data-points related acceleration, load, torque, current, location and material handling events. Real-time data gathered by the edge devices is continuously processed on the edge gateway using dedicated intelligence algorithms.

2) Condition monitoring: Vibrating screens separate material such as rocks according to size. These devices are not complex nor expensive but unexpected failures can cause significant losses in terms of production downtime or disturbances to the processing chain. In these environment condition monitoring is seen mandatory but in terms of precision it ranges from simply knowing something is wrong to identifying exact parts and estimating their condition based on measurements. A predictive system using a digital twin model could tell for example that something is wrong based on a single acceleration measurement sample, whereas long term degradation requires other approaches.

The challenge is that it is not always feasible to transfer all the data to the cloud as costs of networking can grow high, especially in remote locations. An associated complication is the limited access and ownership of data that in some cases is provided from the equipment owner back to the manufacturer through various interfaces. The objective is therefore to simplify the architecture so that processing can take place locally and optimally using edge, fog and cloud resources.

B. Functional Requirements

As functional requirements the system is expected to:

- Acquire heterogeneous measurement data from various sensors and systems and make data streams available
- (Pre)process data locally on the edge and/or in the cloud to balance and optimize computational performance
- Store information in a unified and agreed format for further retrieval and publishing to other data consumers
- Process data aggregates or combination of multiple data streams from various systems and different providers
- Efficiently and centrally manage decentralized information flows and system configurations ensuring information security and dynamic access

In summary, the system should allow managing various information flows and connecting them flexibly to different processing functions and storages, and should offer forward information sharing in a controlled manner to other applications and use cases. As an example, direct access to a manufacturer's own device is not always permitted anymore. A unified access management and orchestration mechanism could mitigate this if the operator would maintain the data itself and would share it forward with e.g. the manufacturer or other service providers.

C. Software and Service Component Interoperability

The system interoperability requirements are considered for data semantics and communication semantics of services. Regarding data interoperability the syntactical messaging formats need to be agreed, e.g. use of structures in JSON or XML. In addition, semantic interoperability means that in order for data to be understandable the individual data values need to have a predefined meaning including e.g. the unit and any other metadata associated with the measurement.

Communication level interoperability means utilizing APIs and communication channels that other system components understand, eventually in a plug and play manner. For typical RESTful APIs this means uniform interfaces and for decoupled approaches (e.g. AMQP or MQTT), it is required that the communication channel structure follows an agreed pattern. Finally, the discovery of services, orchestration and composition, as well as dynamic management of access rights between individual service providers and consumers needs to follow a common and agreed mechanism.

D. Hardware requirements

Setting the service-oriented nature of the system architecture as an objective somewhat limits the support for hardware sensors and systems producing data The requirement set is that they are Internet communication capable (or that a gateway in their substitute is) and can communicate to HTTP REST APIs or to MQTT message brokers that are published and exposed as data sinks for measurement data, i.e. processing or storage.

Regarding run-time environments of the computational functions they should be deployed as services that are agnostic to their physical location. This means that deployment of services needs to be supported so that decoupled functions can be executed on various devices including those of different hardware architectural platforms.

IV. APPLICATION SYSTEM ARCHITECTURE CONCEPT

The architecture concept meeting the previously identified requirements is designed around the Arrowhead Framework [8] core services capable of providing service discovery, orchestration and configuration as well as authorization and access management of services. Figure 1 presents the conceptual architecture outlined in this paper. The example implementation of the architecture concept has been based on sensor measurements provided by Wapice WRM247+ and NodeMCU devices. WRM 247+ devices are robust and highly customisable IoT devices for remote management, measurement and control with built-in sensors, I/O and versatile support for external connectivity and communication protocols. The NodeMCU is a low-cost evaluation board based on the ESP8266 Wi-Fi module. It is Lua based and Arduino compatible with on-board USB-serial, PCB antenna, digital IOs and one analog input.

In the implementation, the measurement devices act as consumers of services that can process and store measurement sensor data. They send information according to an agreed, predefined format to these HTTP REST services which are responsible for processing, analysing or storing the data.

To extend the data exchange alternatives a MQTT channel is offered and the broker serves the MQTT clients and conveys their messages. The MQTT messages can be seen and subscribed by any actor with access rights to the MQTT broker and its specific message topics. This also serves the purpose of concurrently providing a data pipeline, if required, for data based model development which can eventually be deployed locally for improved precision. For enhanced data analytics, the software stack contains a Python3 interpreter.

To promote device agnostic data processing, the information management is built on open standard solution MIMOSA (Machinery Information Management Open System Alliance). MIMOSA is a data model that supports Opens System Architectures for Enterprise Application Integration (OSA-EAI) [12] and for Condition Based Maintenance (OSA-CBM) [13], and it implements ISO-13374-1 [14]. Primary domains are enterprise registry management, condition monitoring, reliability, maintenance and work management functions.

The actual MIMOSA data model is populated into an open source MariaDB database, and can be used at the edge, gateway and cloud levels. The data management and operation event handling is done dynamically in interaction with registered service consumers and providers orchestrated by the Arrowhead Framework. The proposed architecture is device independent, supports interoperability and is flexible to changes. In cases where the local edge level processing is decentralized to data acquisition devices with limited amount of memory, such as Arduino compliant boards (NodeMCU based on ESP8266), the MIMOSA data model is used to combine the edge level run-time data to meaningful events at the gateway level.

In the implementation, CrossControl's CCpilot VS display computer and a CrossLink AI communication module act as gateway devices capable of hosting any computational services part of the service composition including the core services of the Arrowhead Framework.

V. Arrowhead for Dynamic Orchestration, Configuration and Secure Interplay

With service discovery and service composition features of the Arrowhead Framework individual system components, i.e. service consumers and producers, do not need to be configured separately but rather using configurations providing system orchestrations for dynamic and contextual needs. For sensor data analytics services this means that computational functions can be deployed on some computer and executed locally or in the cloud given that the service endpoint is registered to the framework for discovery. With the support of orchestrations consumers and providers are then linked into systems that making use of the authorization core service also ensure that only allowed components can interact with each other.

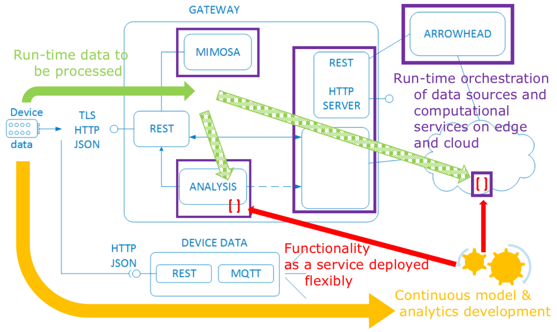

Fig. 1. Conceptual architecture of data flows and computational services.

The orchestration functionality allows managing data flows between components, without having to manage individual services or data sources, when these operate on the Arrowhead Framework model. An example of this is work cycle detection that based on sensor data can calculate and predict workflow steps either locally or in the cloud. This means that decentralized data stream routing and computations can be optimized centrally, e.g. for edge or cloud execution, and decided on any principle (outside of the Arrowhead Framework). The framework can also support in QoS as proposed by [15].

The Arrowhead Framework can be seen as bridge infrastructure integrating different domains. Regarding security it is based on point to point communication between service components and it benefits from tools developed for secure communication between local clouds [16], [17]. Also, the framework does not dictate communication channels, and necessary protocols and means can be used to achieve deterministic real-time communication between individual components.

VI. Conclusion

This paper presented software and communication requirements for a measurement and data analytics system considered for two industrial cases. As a result, an architecture concept for increased flexibility and dynamic configuration of services was designed based on the Arrowhead Framework. Sensor devices send information to services capable of processing and storing data. Using the Arrowhead Framework the configuration is managed dynamically including routing of information and determining where processing takes place to allow optimizing network traffic, edge or cloud computing, or computational precision. The services follow a uniform HTTP REST API which enables the change of service providers making the sensor devices agnostic of the data usage. The framework used verifies the authorization and trustworthiness of system components further easing the burden of individual service consumers and providers in flexible environments.

ACKNOWLEDGMENT

This work is part of the project Productive4.0 and has received funding from the EU ECSEL Joint Undertaking under grant agreement no. 737459 and from the national funding authority Business Finland. Furthermore, the authors are also grateful to the Academy of Finland for their funding supporting this work (grant 310098).

REFERENCES

[1] J. Delsing, *Iot automation: Arrowhead framework*. CRC Press, 2017.

[2] C. Hegedüs, P. Varga, and I. Moldovan, "The mantis architecture for proactive maintenance," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, April 2018, pp. 719–724.

[3] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial big data analytics and cyber-physical systems for future maintenance service innovation," *Procedia CIRP*, vol. 38, pp. 3 – 7, 2015, proceedings of the 4th International Conference on Through-life Engineering Services.

[4] G. Di Orio, P. Mal, J. Barata, M. Albano, and L. L. Ferreira, "Towards a framework for interoperable and interconnected cps-populated systems for proactive maintenance," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, July 2018, pp. 146–151.

[5] E. Jantunen, G. Di Orio, F. Larrinaga, M. Becker, M. Albano, and P. Malo, *A Framework for Maintenance 4.0*, 6 2018.

[6] D. Hästbacka, E. Jantunen, M. Karaila, and L. Barna, "Service-based condition monitoring for cloud-enabled maintenance operations," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 5289–5295.

[7] J. Halme, E. Jantunen, D. Hästbacka, C. Hegeds, P. Varga, M. Björkbom, H. Mesiä, R. More, A. Jaatinen, L. Barna, P. Tuominen, H. Pettinen, M. Elo, and M. Larrañaga, "Monitoring of production processes and the condition of the production equipment through the internet," 2019, 6th International Conference on Control, Decision and Information Technologies, CoDIT'19 ; Conference date: 23-04-2019 Through 26-04-2019.

[8] J. Delsing, "Local cloud internet of things automation: Technology and business model features of distributed internet of things automation solutions," *IEEE Industrial Electronics Magazine*, vol. 11, no. 4, pp. 8–21, Dec 2017.

[9] H. Derhamy, J. Eliasson, and J. Delsing, "System of system composition based on decentralized service-oriented architecture," *IEEE Systems Journal*, pp. 1–12, 2019.

[10] F. Montori, T. S. Cinotti, and D. Brunelli, "Smart energy services integrated within the arrowhead communication framework," in *2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, June 2016, pp. 558–563.

[11] A. Zabasta, K. Kondratjevs, J. Peksa, and N. Kunicina, "Mqtt enabled service broker for implementation arrowhead core systems for automation of control of utility' systems," in *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Nov 2017, pp. 1–6.

[12] *OSA-EAI Open System Architecture for Enterprise Application Integration V3.2.3 Production Specification*, MIMOSA, 2012.

[13] *OSA-CBM Open System Architecture for Condition-Based Maintenance V3.3.1 Production Specification*, MIMOSA, 2010.

[14] *ISO 13374-1:2003: condition monitoring and diagnostics of machines data processing, communication and presentation part 1: General guidelines*, 03 2003.

[15] M. Albano, P. M. Barbosa, J. Silva, R. Duarte, L. L. Ferreira, and J. Delsing, "Quality of service on the arrowhead framework," in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, May 2017, pp. 1–8.

[16] C. Hegedus, D. Kozma, G. Soos, and P. Varga, "Enhancements of the arrowhead framework to refine inter-cloud service interactions," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 5259–5264.

[17] C. Hegedus, P. Varga, and A. Frank, "Secure and trusted inter-cloud communications in the arrowhead framework," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, May 2018, pp. 755–760.