Contents lists available at ScienceDirect

CIRP Annals
Manufacturing Technology
Journal homepage: www.elsevier.com/locate/cirp

Scalable Data Pipeline Architecture to Support the Industrial Internet of Things

Moneer Helu (3)[a], Timothy Sprock[a], Daniel Hartenstine[b], Rishabh Venketesh[a], William Sobel[c]

[a]Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA
[b]Millersville University, Millersville, PA 17551 USA
[c]VIMANA, Oakland, CA 94612 USA

Submitted by Robert G. Wilhelm (1), University of Nebraska, Lincoln, NE 68588 USA

Managing manufacturing data remains challenging despite the growth of the Industrial Internet of Things (IIoT). While various standards and technologies enable greater access to data, scaling data processing and distribution can be difficult given the increasing variety of data from an increasing variety of sources in global production networks. This paper proposes an architecture for a scalable pipeline to process and distribute data from a mix of shop-floor sources. The feasibility of this approach is explored by implementing the architecture to bring together MTConnect-compliant machine and ad-hoc power data to support analytics applications.

System architecture, Standardization, Data

1. Introduction

The Industrial Internet of Things (IIoT) concept (or the Industrial Internet) describes a network of objects in an industrial environment (e.g., machine tools) that enables information sharing, interaction, and collaboration between the things themselves [1-3]. It provides an infrastructure for the related concepts of Smart Manufacturing, Industry 4.0, and Cyber-Physical Systems (CPSs) by allowing data collection and distribution for data-driven applications that support decision making and control [3-6]. Such applications are essential in helping manufacturers address the challenges associated with increasingly distributed global production systems [5,7,8]. However, the increasing variety of data and number of systems in a global production environment that support IIoT applications requires an architectural approach where the management of data does not presuppose its use so that this data can be used many applications to maximize value.

Generating value from IIoT data requires transforming raw data to semantically-rich data that can be curated to create the context needed for different viewpoints [9]. This process starts on the shop floor by collecting data from devices using Application Programming Interfaces (APIs) and data access protocols (e.g., EtherNet/IP, Modbus). Standards can be used to normalize, classify, and contextualize data in a consistent and interoperable way across many devices and applications. For example, MTConnect (ANSI/MTC1.5-2019) enables semantic interoperability by defining a vocabulary for manufacturing systems to provide structured, contextualized data [10].

Solutions to transform, process, and organize data have evolved quickly in many domains as industries have matured to take advantage of the larger Internet of Things (IoT) concept. As the IIoT matures, global production networks may benefit from these solutions, but research is needed to develop appropriate architectures that help all manufacturers select, configure, and deploy proven data transport and processing solutions developed in non-manufacturing domains. Such work enables shop-floor ecosystems where new data sources and consumers can be easily plugged in, data can be processed and distributed at scale, and constraints of operational environments with existing heterogeneous technologies can be respected.

This paper proposes an architecture to address shop floor connectivity and distributed data management concerns using enterprise-grade middleware. To do so, we divide the data pipeline into two functional components that (1) collect and move data away from the shop floor and (2) scale the distribution of that data. Finally, we explore the feasibility of this approach by using it to bring together MTConnect-compliant and ad-hoc power data to support analytics applications.

2. Background

Figure 1 shows how to transform IIoT data to maximize its value across multiple applications. We give structure and meaning to collected raw data using standardized domain information models. The data can be curated for different viewpoints before distribution to an application ecosystem where it can be integrated with other production systems or other purposes (e.g., analytics).

When deploying new technologies for this data transformation pipeline into legacy environments (e.g., the shop floor), system architects must maintain existing interfaces while improving functionality at scale (i.e., support more data from more

sources for more clients). Developing such solutions requires that we first assess the sources (i.e., MTConnect-compliant devices) and clients (i.e., analytics applications) for the data pipeline.

2.1. Typical MTConnect Implementation

MTConnect provides a standard domain model that ascribes meaning to data from shop-floor equipment through a controlled vocabulary, typing system, and relationships between data elements [10]. The standard also describes minimum infrastructure to access and transform data collected from shop-floor equipment through a uniform interface. Figure 1 shows the architecture of this infrastructure. The *Adapter* interfaces with the equipment API and works with the *Agent* to translate data from proprietary structures and representations to the MTConnect controlled vocabularies, units, and types. The Agent also provides metadata contextualization, relationships between the elements, and formatting of the data into response documents for client requests. These requests are handled through a lightweight Representational State Transfer (REST)ful interface. Alternatively, the MTConnect-OPC Unified Architecture (UA) Companion Specification describes methods for providing MTConnect semantics with the OPC-UA information model and protocol [11].

Many MTConnect-based ecosystems start by collecting data from Agents and storing this data in a database to serve to applications or filtering or synthesizing the data stream into a dashboard and discarding the excess data. In a semi-decentralized application ecosystem where there is no single central data repository, each application collects the data it needs from a set of Agents. This data does not need to be complete nor does the application need to store it any longer than it requires.

2.2. Analytics Applications in Manufacturing

Analytics is the process of transforming data into actionable information using systematic analysis [12]. Analytics in manufacturing (or Industrial Analytics) has been extensively reviewed [13-16]. Industrial analytics applications include prognosis and Prognostics and Health Management (PHM) [3,4], management and control of production systems [8,17,18] and CPS [6], and human-robot collaboration [19]. Despite increasing interest in industrial analytics, most literature focuses on applying analysis methods to relatively large datasets pre-stored in curated databases rather than the timely transformation and distribution of streaming data for multiple applications [20,21].

The growth of industrial analytics has lagged developments in other data-rich domains, such as finance and online platforms. In these domains, data management has evolved from database-centric (e.g., see Section 2.1) to distributed data-science pipelines [12,22]. While the fundamental objectives of these approaches have remained consistent (i.e., derive insight from data), system architectures have had to handle larger quantities of streaming data with acceptable latency response times. So, lambda architectures have become a common pattern of most analytical systems, including some used in manufacturing [23,24].

The lambda architecture has two layers of parallel information flows: (1) speed and (2) batch [22]. Components in the speed layer continuously process events and use trained machine learning models to classify and react to events in industrial processes. Speed systems must provide feedback before the analysis results become irrelevant and value is lost. Conversely, batch systems provide a periodic or on-demand analysis using historical data to create and refine learning models or perform more complex contextual

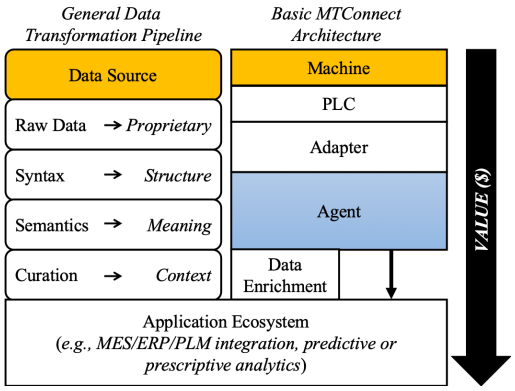

Figure 1. General data transformation process mapped to MTConnect

analysis where the resulting latency of the insights is not as time-critical as the speed systems. This separation of concerns allows each system to better address its objectives, which is why a lambda pattern has been suggested for IIoT applications [12].

2.3. Architectural Concerns

Industrial analytics typically assumes that data is readily-consumable, error-free, and traceable to its source. However, we intuitively understand that this is not true, which drives the primary architectural concerns underlying data transformation. While compute and storage technologies evolve quickly, these changes should not significantly affect the architectural concerns derived from the requirements of the use case. Important architectural concerns for transforming data from the shop floor include

- Format data into standardized data types,
- Synchronize timestamps to assert causality of events from multiple sources,
- Use Original Equipment Manufacturer (OEM) knowledge to translate programmable logic controller (PLC) tags and units into standardized controlled vocabularies and data formats,
- Enable deployment on embedded or legacy systems,
- Manage continuity of data stream and recovery,
- Ensure data provenance can be asserted by applications,
- Ensure security of equipment from external incursions,
- Route information flows to multiple endpoints, and
- Enable data persistence in a permanent immutable store.

A key concern in manufacturing is the mix of Information Technology (IT) and Operational Technology (OT) systems on the shop floor. When selecting technologies to address each architectural concern, the placement of these solutions in IT or OT systems requires trade-offs between the criticality of response time and the ease of maintenance since solutions become harder to patch and maintain once you cross the OT boundary. When data is collected for less time-sensitive applications, such as operations management functions, IT solutions for routing data to multiple endpoints may better address the following additional concerns, which are in the focus of the middleware described in this paper:

- Merge heterogeneous data streams, including MTConnect-compliant and other ad-hoc data sources;
- Ensure scalability and elasticity with respect to memory, processing power, and bandwidth;
- Maintain existing or comparable interfaces at scale; and
- Provide high availability of critical components.

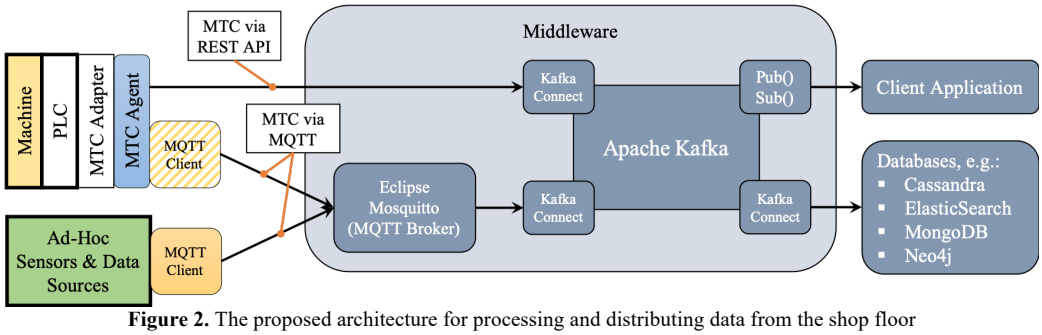

Figure 2. The proposed architecture for processing and distributing data from the shop floor

3. Proposed Architecture

The proposed architecture (Figure 2) has been designed to address the concerns identified in Section 2.3. This architecture leverages middleware, which is an architectural pattern that integrates hardware and software components to support communication between distributed systems. It provides a set of design attributes realized using the elements described in this section to enable two separate functions: moving data from the shop floor and distributing this data at scale.

3.1. Moving Data from the Shop Floor

In ecosystems of distributed applications, each application is responsible for requesting the data they need, leading to the same data being requested repeatedly. Because MTConnect Agents are often deployed on the machine (i.e., an OT system), it can be challenging to scale the Agent's capacity to respond to large or complex data requests from multiple applications. Offloading this functionality to a dedicated message broker enables the data service to be scaled with available resources rather than constrained to the capacity of legacy hardware with limited memory and computing resources.

Message Queuing Telemetry Transport (MQTT) (ISO/IEC 20922) is a publish-subscribe network protocol that transports messages between devices [25]. It is lightweight with respect to processing, bandwidth, and power consumption, and it supports connections over Transport Layer Security (TLS) for security. Like many message brokers, MQTT brokers offer Quality of Service (QoS) levels so that, when required, they can guarantee that any subscriber will receive every piece of data published from a machine. Other protocols can provide similar functionality, but we prioritize lightweight, easily-deployable solutions to move data from the shop floor.

One substantial limitation to message brokers on the shop floor is their limited short-term retention. Message brokers delete messages after all subscribers have received their data. In cases where there are no subscribers to a topic, the message may be discarded immediately, which presents a challenge when applications are instantiated intermittently to answer questions since they will not have access to recent data. Addressing this challenges requires dedicated technology for storing, scaling, and distributing data.

3.2. Distributing Data at Scale

Apache Kafka is a high-throughput, low-latency software platform for streaming data with infrastructure designed for efficiently consuming, storing, recalling, and producing data. Its ability to consume and produce data efficiently relies on defining units of work called *tasks*, e.g., "get data from MTConnect agent" or "write values to relational database." Given a set of heterogeneous tasks, Kafka organizes the task execution to complete tasks as quickly as possible. Kafka stores the resulting data in an append-only, immutable log store until the server runs out of space or per retention policy. Client applications that get disconnected or new application subscribers can recreate a significant portion of the recent data stream history, which allows applications to trust that they are accessing the original data.

The architecture of Kafka's interface and data storage enables servers and memory to be added dynamically to support additional workload. Other features, such as immutable logs and strong replication across data storage partitions, enable robust, traceable shop-floor data collection. Thus, Kafka provides a dedicated technology to hold data, respond to requests, and scale to meet storage and processing demands so that IIoT data can be made accessible to many applications that each use this data differently.

4. Implementation

The architecture shown in Figure 2 is based on (1) Eclipse Mosquitto (an MQTT broker), (2) Apache Kafka (including Apache Kafka Connect) to collect data from the REST API of MTConnect-compliant sources, (3) MongoDB (a NoSQL database) to provide persistent data storage, and (4) Docker (a container service) to simplify deployment. Many of these components can be found in deployable containers and represent commercial-off-the-shelf (COTS) tool that simply need to be instantiated, configured, and connected together. Apache Kafka and its accompanying Kafka Connect API enable easy connections to upstream and downstream data producers and consumers. For example, we used existing Kakfa connectors to deploy an Eclipse Mosquitto broker to publish to Apache Kafka and MongoDB to consume from Kafka. In each case, the configuration specifies topics that messages will be produced to or consumed from.

Our work first focused on connecting generic data pipeline components to shop-floor data sources, specifically, a Hurco VMX24 machine tool in the National Institute of Standards and Technology (NIST) Smart Manufacturing Systems (SMS) Test Bed. Traditional approaches gather data from MTConnect Agents using Hypertext Transfer Protocol (HTTP) requests. As a first step, we integrated the traditional approach with proposed pipeline by publishing the response document to the Mosquitto broker rather than writing it directly to a database. We next developed a Kafka Connector that structured the HTTP requests into tasks. When executed, each task requests an interval of data from the Agent, stores the sequence number (i.e., index) of the last collected item, and puts the data into a Kafka topic (see "MTC via REST API" in Figure 2). Given a list of devices, Kafka optimizes its use of system resources to collect data from those devices.

The second portion of our research focused on integrating a non-MTConnect compliant data source into the same pipeline. A

Beckhoff PLC was used to collect power data from the Hurco VMX24 and publish the data using MQTT (see "MTC via MQTT" in Figure 2). The data was organized in the MQTT broker under the same parent topic used for the MTConnect-compliant data collected from the Hurco, allowing it to be requested as a part of the Hurco's data. The same storage approach was used in Apache Kafka. Using our proposed pipeline, the power data from the Beckhoff PLC was brought into the same environment as the MTConnect-compliant data from the Hurco VMX24 so that it could then be aligned and synchronized to support further analysis. This approach reflects the need to identify, store, and transport the data identically regardless of its source. This demonstration also evaluated the feasibility of putting an MQTT client in the MTConnect Agent itself, potentially enabling MTConnect data be published natively as MQTT messages.

5. Summary

The proposed architecture and implementation accommodate shop-floor use cases for processing and distributing data from a mix of sensors and devices. It provides a framework to explore other implementations for collecting and serving MTConnect-compliant data supported by an IIoT ecosystem of dedicated applications providing other functions. Data streaming from these shop-floor devices can be processed during the streaming or from persistent storage (i.e., MongoDB). Apache Kafka and the Apache Foundation's "Big Data" ecosystem contain tools that can be connected to the Kafka backbone, implementing variants on the lambda architecture.

While the maturity of shop-floor connectivity and distributed data management solutions has lagged other data-rich domains, adapting and deploying proven architectures and technologies from these industries is an effective solution. For example, the performance of proposed components, in terms of reliability, latency, and throughput, has been proven in demanding environments. Industrial state of art is streaming analytics platforms supported by lambda architectures that analyze and transform data before storing results in long-term storage.

The proposed data pipeline architecture satisfies integration requirements by leveraging widely supported technologies to flexibly gather information from heterogenous sources, not limiting the solution to connecting only MTConnect-based ecosystems. Both the Asset Administration Shell of RAMI4.0 and OPC-UA provide guidance on deploying MQTT as part of the architecture to transport data from the shop floor to data processing and storage capabilities. These sources of information and other IoT devices can be connected seamlessly, via MQTT, to the data pipeline proposed in this paper. Information standards may vary.

While the technologies selected to implement the architecture have been proven in non-manufacturing environments; additional research is needed to quantify the ability of the proposed architecture to scale across hundreds to thousands of manufacturing data sources. To do so, these data sources can be simulated in a computer-cluster test environment to assess the latency response times that may be achieved. Additional high-bandwidth data sources can also be incorporated into the pipeline to explore the amount of data that may be successfully managed. Finally, there are opportunities to connect the proposed architecture to on-going industrial analytics research exploring data alignment and post-processing requirements. Such research may help identify other technologies that can be integrated into this implementation so that more applications can leverage the growing availability of IIoT data from the shop floor.

Disclaimer

References

[1] Evans, P. C., Annunziata, M., 2012, Industrial Internet: Pushing the Boundaries, General Electric.
[2] Atzori, L., Iera, A., Morabito, G., 2010, The Internet of Things: A Survey, Computer Networks, 54/15:2787-2805.
[3] Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., Eschert, T., 2017, Industrial Internet of Things and Cyber Manufacturing Systems, in Industrial Internet of Things, Springer, 3-19.
[4] Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., Helu, M., 2015, Cloud-Enabled Prognosis for Manufacturing, Annals of the CIRP, 64/2:749-772.
[5] Helu, M., Hedberg T., Feeney, A. B., 2017, Reference Architecture to Integrate Heterogeneous Manufacturing Systems for the Digital Thread, CIRP J. Manufacturing Science and Technology, 63/2:585-605.
[6] Monostori, L., Kádár, Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., 2016, Cyber-Physical Systems in Manufacturing, Annals of the CIRP, 65/2:621-641.
[7] Hedberg, T., Helu, M., Sprock, T., 2018, A Standards and Technology Roadmap for Scalable Distributed Manufacturing Systems, Proc. ASME 2018 Manufacturing Science and Engineering Conf., V003T02A019.
[8] Lanza, G., Ferdows, K., Kara, S., Mourtzis, D., Schuh, G., Váncza, J., Wang, L., Wiendahl, H.-P., 2019, Global Production Networks: Design and Operation, Annals of the CIRP, 68/2:823-841.
[9] Hedberg, T., Feeney, A. B., Helu, M., Camelio, J. A., 2017, Toward a Lifecycle Information Framework and Technology in Manufacturing, J. Computing and Information Science in Engineering, 17/2:021010.
[10] ANSI MTConnect v1.5.0, 2019, MTConnect Institute.
[11] OPC Unified Architecture for MTConnect (OPC 30070-1) v2.00, 2019, MTConnect Institute and OPC Foundation.
[12] Anderson, N., Diab, W. W., French, T., Harper, K. E., Lin, S.-W., Nair, D., Sobel, W., 2017, The Industrial Internet of Things Volume T3: Analytics Framework, Industrial Internet Consortium, IIC:PUB:T3:V1.00:PB:20171023.
[13] Li, B.-H., Hou, B.-C., Yu, W.-T., Lu, X.-B., Yang, C.-W, 2017, Applications of Artificial Intelligence in Intelligent Manufacturing: A Review, Frontiers of Information Technology and Electronic Engineering, 18/1:86-96.
[14] Sharp, M., Ak, R., Hedberg, T., 2018, A Survey of the Advancing Use and Development of Machine Learning in Smart Manufacturing, J. Manufacturing Systems, 48:170-179.
[15] Wuest, T., Weimer, D., Irgens, C., Thoben, K.-D., 2016, Machine Learning in Manufacturing: Advantages, Challenges, and Applications, Production and Manufacturing Research, 4/1:23-45.
[16] Zhong, R. Y., Xu, X., Klotz, E., Newman, S. T., 2017, Intelligent Manufacturing in the Context of Industry 4.0: A Review, Engineering, 3/5:616-630.
[17] Frazzon, E. M., Kück, M., Freitag, M., 2018, Data-Driven Production Control for Complex and Dynamic Manufacturing Systems, Annals of the CIRP, 67/1:515-518.
[18] Gödri, I., Kardos, C., Pfeiffer, A., Váncza, J., 2019, Data Analytics-Based Decision Support Workflow for High-Mix Low-Volume Production Systems, Annals of the CIRP, 68/1:471-474.
[19] Wang, L., Gao, R., Váncza, J., Krüger, J., Wang, X. V., Makris, S., Chryssolouris, G., 2019, Symbiotic Human-Robot Collaborative Assembly, Annals of the CIRP, 68/2:701-726.
[20] Mourtzis, D., Vlachou, E., Milas, N., 2016, Industrial Big Data as a Result of IoT Adoption in Manufacturing, Procedia CIRP, 55:290-295.
[21] Ritou, M., Belkadi, F., Yahouni, Z., Cunha, C. D., Laroche, F., Furet, B., 2019, Knowledge-Based Multi-Level Aggregation for Decision Aid in the Machining Industry, Annals of the CIRP, 68/1:475-478.
[22] Marz, N., Warren, J., 2015, Big Data: Principles and Best Practices of Scalable Real-Time Data Systems, Manning Publications Co.
[23] Rix, M., Kujat, B., Meisen, T., Jeschke, S., 2016, An Agile Information Processing Framework for High Pressure Die Casting Applications in Modern Manufacturing Systems, Procedia CIRP, 41:1084-1089.
[24] Yamato, Y., Kumazaki, H., Fukumoto, Y., 2016, Proposal of Lambda Architecture Adoption for Real Time Predictive Maintenance, Proc. IEEE 2016 Intl. Symp. On Computing and Networking, 713-715.
[25] Information Technology – Message Queuing Telemetry Transport (MQTT) v3.1.1 (ISO/IEC 20922:2016), 2016, International Organization for Standardization and International Electrotechnical Commission.