Monitoring and Control of Unstructured Manufacturing Big Data

Yesheng Cui[1], Sami Kara[1], Ka C. Chan[1,2]
1 School of Mechanical and Manufacturing Engineering, The University of New South Wales, Sydney, Australia
2 School of Management and Enterprise, University of Southern Queensland, Springfield, Australia
(yesheng.cui@student.unsw.edu.au)

***Abstract* - Unstructured manufacturing big data silos are challenging for enabling various data-driven applications such as digital threads and digital twins in manufacturing. The management of big data silos requires to address the issues of large volume, data inconsistency, data redundancy, information silos and data security. This research developed a systematic approach to managing data silos using the state of art big data software. Applying this approach in the product life cycle can control data silos, data consistency, redundancy, timely update and enable the automatic workflow of each system.**

***Keywords* - Big data, unstructured data, Industry 4.0,**

I. INTRODUCTION

Data is essential to achieve Industry 4.0 and smart manufacturing. One feature of Industry 4.0 is the end-to-end engineering digital integration of the entire product value chain from product design to product development, manufacturing engineering, production and service [1]. Data is the basic medium for end-to-end digital integration, because data is generated in each phase and exchanged to integrate with the other phases. By collecting and analysing big data at all phases of the product value chain, manufacturing can be more efficient and smarter [2].

The data processed and exchanged at each phase has a different structure, format and velocity. As manufacturing applications are extensively digitalised, each phase of the product value chain uses various software and generates various data, including CAD, CAE, CAPP, CAM, CNC, QC etc. In the product design phase, Computer Aided Design (CAD) and Finite Element Analysis (FEA) software are used to assist designers to generate product models and analyse product weaknesses before product manufacturing, thereby generating unstructured data with different file extensions such as STL, IGES etc. In the product manufacturing phase, Computer-Aided Manufacturing (CAM) is used to control machine tools for faster and more efficient production process, thereby generating text files with standard codes such as G-codes/M-codes [3]. Machine tools use XML formats such as MTConnect to generate process information from massive streaming time-series data [4]. Product model information (PMI) can be transformed from unstructured files to structured spreadsheets with product parts lists, which can be used for analysis and reporting of any semantic PMI [5]. Moreover, as the division of work in each phase becomes clearer and dedicated, different people operate the software to generate various data stored in different locations on different computers, and become siloed data.

The data silo problem is challenging for the implementation of Industry 4.0 and smart manufacturing, such as end-to-end digital integration of product value chains, digital twins and digital threads, which require data collection and transmission, sharing with the systems in the product lifecycle and supply chain [4]. An overview of the technical approach of the digital thread of the three systems (CAD, CAM, CNC) in the product dimension is shown in Fig. 1. The data collected by the three systems can be parsed and analysed to provide more optimal decision making. However, it does not provide a solution to the collection of siloed data. Moreover, data collection is still a challenging issue for implementing data-driven smart manufacturing [6]. Hence, addressing the data silo problem is needed in order to realise the digital thread concept.

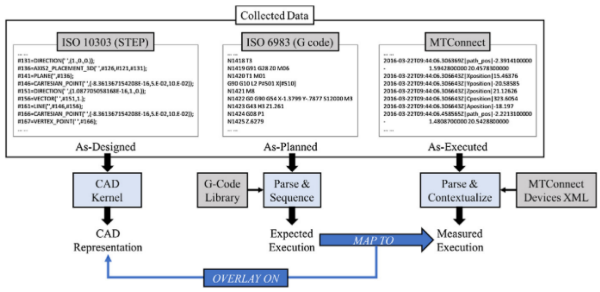


Fig. 1. Overview of technical approach to link as-planned to as-executed data and overlay results on CAD representation [3]

To solve the data silo problem, the following issues should be considered, including timely monitoring of data changes, data redundancy, data inconsistency and data insecurity. Firstly, the siloed data cannot be found easily. This means that data users may have to search different directories on multiple computers and cloud repositories to find the required data. Secondly, although the data in the upstream phase can be sent to the data users in the downstream phase, it leads to the data redundancy issue, that is, the same data is redundantly stored in different locations, wasting data storage space. In addition, the data users in the downstream phase cannot track the latest redundant data in time, since the data in each phase is generated at a different velocity. This can cause unnecessary delays and inefficiencies in the entire workflow. Thirdly, the data silo problem may lead to inconsistent data since the data in each phase have different formats and versions. For example, a product is designed using one CAD file in the product design phase but it is converted to two CAM files which correspond to two different machines or operations in the product manufacturing phase. Before using the CAM files for manufacturing, engineers or machinists face the challenge of identifying inconsistent data. Furthermore, the data silo problem increases the risk of data leakage. Because product data is redundantly stored in different places, anyone can access most of the data by accessing any of the data repositories.

The current solutions conceptually and theoretically addresses the data silo problem in manufacturing [4], without considering the discussed data issues [3]. In this research, several big data software was selected to verify the proposed methodology to supervise unstructured data in the product dimension. It addresses the data silo problem by providing a uniform platform to integrate siloed data from disperse systems, capture changes of siloed data in a timely manner, store data persistently, monitor data redundancy and data inconsistency, automate workflow and enable the concept of digital thread.

We propose a solution called Data Control Module (DCM) to address these issues. The rest of this paper is presented as follows: Section 2 introduces the methodology and the big data software used; Section 3 describes the design and setup of the system; Section 4 discusses the experimental results; and Section 5 presents the conclusion of this paper.

II. METHODOLOGY

A. DCM architecture

The DCM architecture in the product dimension is shown in Fig. 2. The architecture consists of DCM Cloud and a number of DCM Edge systems. DCM Cloud communicates with DCM Edge systems via an IP network.

At the edge, there are three data producers and consumers. These are computers on which CAD, CAM and CNC software is installed separately. Different people operate different software to generate data and send the data to people in the downstream system to use. DCM Edge addresses the issue of timely monitoring the latest data on the data silos. DCM Edge is deployed in each of the edge computer to monitor the latest generated data in a timely manner, collect and send the data with corresponding metadata to DCM Cloud, and execute the control message functions of DCM Cloud, such as getting data from DCM Cloud to the expected location. CNC machines generate streaming XML data (MTConnect) which is semi-structured, and data collection of CNC machines data was published in a previous paper [6]. The focus of this paper is on unstructured data.

On the cloud side, DCM Cloud focuses on addressing integration of siloed data for analysis or other applications. Data management, visualisation and security address the issues of data redundancy, data inconsistency and insecurity. The specific functions of the components are as follows:

- Data collection collects raw data and metadata from DCM Edge and loads them into the data lake and data warehouse. Data transaction processing formats of metadata into a consistent format for query and analysis.
- Data lake integrates the unstructured data with the raw format. Since metadata has common features, such as data locations on the edge computers and DCM Cloud, the computer name, data owner, created date and modified date, data size and file extension, the metadata of the collected data should be formatted into structure data. The common features can be used to quickly find data and analyse metadata. For example, if people want to find a file without knowing the file name, they can identify the file by searching the relevant information from the metadata table such as file extension, file created date, owner etc. If DCM identifies redundant data in the data lake, manufacturers can identify data redundancy by summing up the size of the redundant data from the metadata table. Hence, the metadata is stored in tabular format in the HBase database of DCM Cloud.
- Data control implements the created rules to control data of the location of specific data silos to the expected location of data silos. It can automate the workflow among CAD, CAM and CNC DCM Edges.
- Data management addresses data redundancy and data inconsistency through data redundancy management and data provenance.
- Data users use data visualisation to identify consistent data by viewing the content of raw data in DCM Cloud.
- Security protects data in DCM Cloud from unauthentic and unauthorised access.

B. Big data software

As discussed in the previous section, big data software is selected to implement the functions of the components of the DCM architecture. In addition, big data software provides horizontal scalability which means clusters of computers can operate to store and compute data on DCM Cloud. As data accumulates over time, DCM Cloud requires more storage and computation resources to store and process data. Big data software can add new computers to the existing clusters of DCM Cloud without upgrading the computer hardware [7].

- Apache NiFi

Apache NiFi is a data flow big data software that can automatically implement various functions to operate data [6]. Based on the developed DCM architecture, NiFi was selected as a component of monitor and executor at DCM Edge; and a component of data collection, allocation and ETL functions at DCM Cloud. Moreover, NiFi has a data provenance function that can trace the history of data and can be used to check data consistency.

- Apache HDFS

HDFS was selected as the data lake of DCM Cloud. It can store the collected unstructured data in raw format persistently and redundantly [9]. If the computer of DCM Cloud fails, the data in DCM Cloud will not be lost since the data is redundant in other computers. At the same time, data users will not observe any difference to view and operate data.

- Apache HBase

HBase is column-based NoSQL database which can store structured and unstructured data. As mentioned above, the metadata is preferably formatted in tabular format.

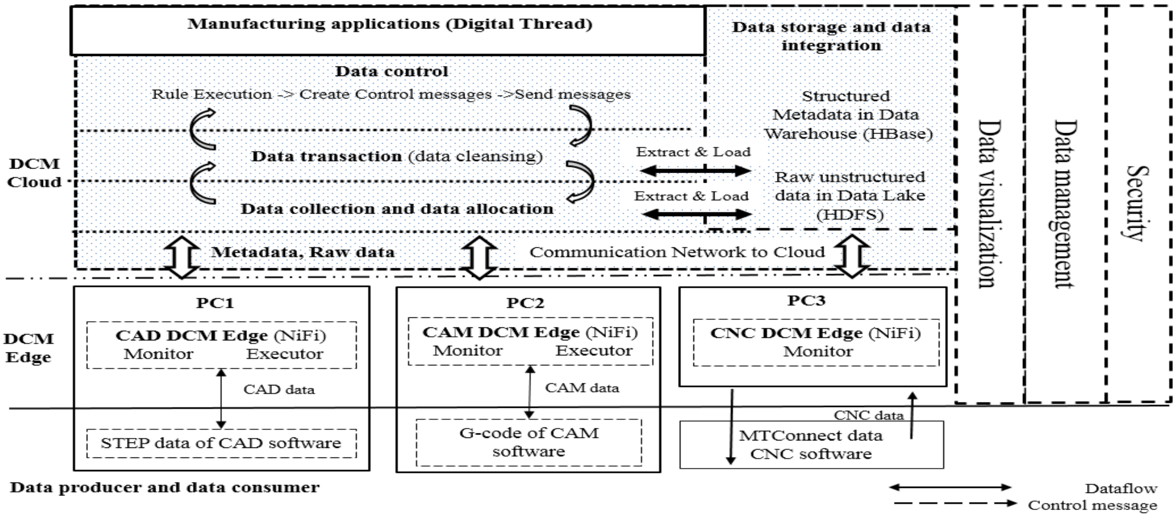


Fig. 2. Digital thread application with DCM methodology

The common features of metadata are defined as the columns of the table in HBase. However, the features of metadata are not fixed. Some users may add new features to the metadata of the data such as product design files of specific batches. Hence, new features must be added as new columns to existing metadata tables. If relational database is used, a new table must be re-created since the schema of the relational database cannot be modified once it is created. Since HBase is NoSQL, new columns can be added to existing tables without affecting usage.

- Apache Phoenix

Apache Phoenix is the SQL skin of HBase. Phoenix allows users to query HBase data using SQL without duplicating the data from HBase to a SQL database. Because HBase is not a SQL database, it uses its own query language to query metadata. However, SQL is familiar and convenient for users to analyse structured metadata. Therefore, Phoenix has been added to HBase.

- Apache Kafka

Kafka is designed for high-throughput, low-latency messaging platform [8]. Kafka is suitable for transmitting control messages between DCM Cloud and DCM Edge. Control messages are defined as topics which include data information such as source location and data source computer, expected location and data computer etc. Because there are many data users in manufacturing who create many rules to control various data, the messaging software should support the transmission of massive messages. Moreover, data users may create more new rules over time. If the software cannot support horizontal scalability, the newly created messages will affect the performance of the control message transmission. Moreover, Kafka has a maximum file size (1MB) limit, suitable for sending control message. Therefore, Kafka is an ideal candidate for transmitting control messages between DCM Edge and DCM Cloud.

III. SYSTEM DESIGN AND SETUP

This section describes the system design, setup, and configuration of each DCM component.

A. Experimental setup

Fig. 3. shows the experimental setup of the DCM system. Based on the architecture, three computers were used as the data consumers and producers, and the DCM Cloud. Three virtual machines (VM) were created in the computer PC3 with the Linux operating system installed. DCM Edge was installed on each of the three VMs, and an open-source manufacturing software (FreeCAD, CAMotics and LinuxCNC) was installed on each of the three VMs. Different IP addresses have been allocated to the three VMs to simulate the real working environment in manufacturing. The computer PC1 implements functions including data collection and allocation, data transaction and data provenance. The computer PC2 was used for data storage, rule-based data control, data redundancy management, and data visualisation.

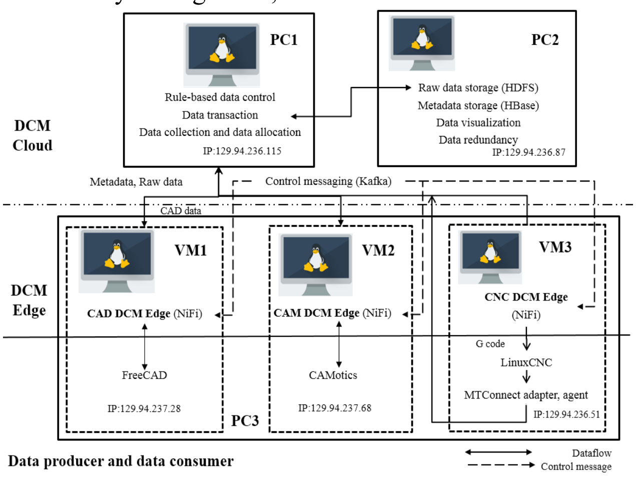


Fig. 3. The experiment setup of the designed system

B. DCM Edge

Based on the DCM Edge requirements, NiFi was selected for the development of DCM Edge. Fig. 4. shows the NiFi flowcharts of the Monitor and Executor of the DCM Edge. These systems will be described in this section.

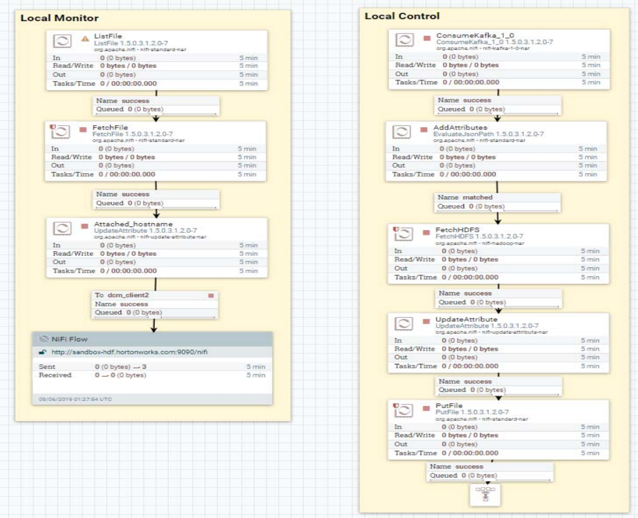


Fig. 4. NiFi Data flow of Monitor and Executor of DCM Edge

C. DCM Cloud

Based on the analysed requirements of DCM Cloud, the developed DCM Cloud is divided into two parts: one part is data collection and data allocation, and data transaction to format metadata; and the other part is data storage for raw unstructured data and metadata. DCM Cloud uses a consistent method to collect data from various DCM Edges, stores the data in raw format in the universal directory, records the metadata with the consistent information from both views of edge and cloud. For example, if the names of data collected from different data silos are the same, a name conflict error will occur when saving the data in the DCM Cloud. DCM Cloud classifies data according to the metadata information and identifies data redundancy by comparing duplicate data on DCM Cloud. Data is controlled with the created rules. Fig. 5 shows the flowchart of the developed DCM Cloud. In the top left corner, DCM Cloud implement NiFi to collect and process raw data and the corresponding metadata from three DCM Edge systems . Raw data is renamed with a unique ID number (global UUID) by consistent naming methods. Using the specific NiFi processors to format metadata of each raw data to JSON and store it in the metadata table of HBase.

In the bottom right corner, DCM Cloud implements HDFS and HBase to store raw data and metadata. Data redundancy management and data control are implemented by developed Python programs. Redundant data is identified by comparing raw data on HDFS and store results in the redundancy table of HBase. Data control rules make the requested siloed data on one edge system available on the request edge system. Therefore, the data location information on source and target edge systems have to be embedded in the created Data control rules. Data allocation of DCM Cloud implements the rules to fetch the requested data from HDFS and locate it on the destination of target edge systems. .

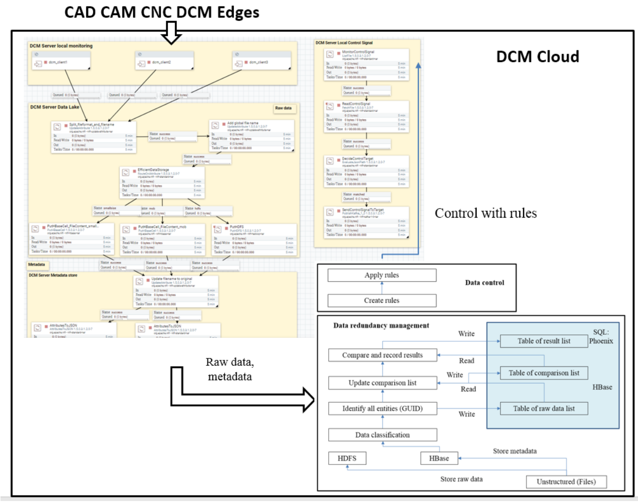


Fig. 5. The flowchart of the developed DCM Cloud

1. Data storage and data integration

- Physical and logical data storage

Raw unstructured data is stored in HDFS in its raw format. Related metadata is stored in the WORKTABLE of HBase in a structured format.

- Physical and logical data integration

For the purpose of physical data integration, the unstructured raw data is stored under the universal file directory of HDFS in the DCM Cloud.

- Search and query

The raw data stored in HDFS can be found by searching the global UUID of the raw data or the local filename. The complete metadata information of the raw data can be queried using SQL code through the Phoenix interface.

2. Data control

Data control rules are created and control messages are sent to specific DCM Edge systems. The name of the DCM Edge is used as the topic of Kafka to ensure that the messages of the Kafka publisher of DCM Cloud is sent to and retrieved by the Kafka consumer of DCM Edge.

3. Data redundancy management

The flowchart in the lower right corner of Fig. 5 illustrates the process of data storage, data redundancy management and data control of DCM Cloud with unstructured data. Data redundancy management classifies the collected data by file extension of the metadata table and stores the classified data list in a different table of HBase. Then, it generates a comparison list from the classified data table to compare the data one by one. The developed Python program compares the raw data in the list, which is retrieved from HDFS of DCM Cloud, and records the results in the result table.

IV. EXPERIMENTAL RESULTS

This section presents the experimental results of implementing the developed DCM Edge and DCM Cloud.

A. Data control

Fig. 6 shows the data executors of two DCM Edges. The unstructured data (the green file) is sent from a data producer to a data consumer.

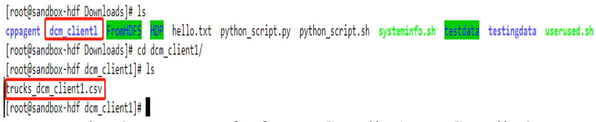


Fig. 6. Data transfer from DCM silo 1 to DCM silo 2

B. Data storage and data integration

Fig. 7 shows that all the collected unstructured data is stored in the raw format in HDFS of the developed DCM Cloud. The structure of the data filename has two parts, first the file extension, second the global UUID generated from Apache NiFi as the unique ID to identify each data.

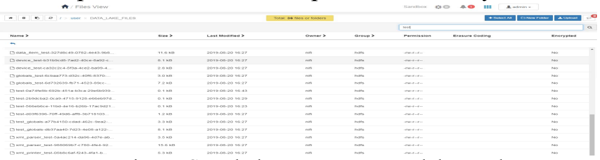


Fig. 7. Search the raw unstructured data and

C. Data management

Fig. 8 shows that the raw unstructured data stored in HDFS can be viewed by searching the global UUID or the local filename.

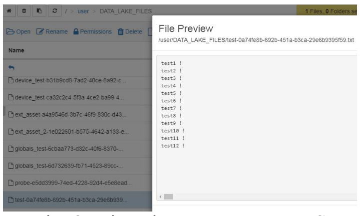
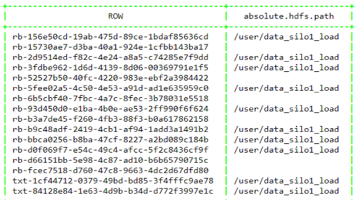


Fig. 8. view the content on HDFS

Fig. 9 shows that the metadata of the raw unstructured data is stored in an unstructured format in HBase.



ROW	absolute.hdfs.path
rb-156e50cd-19ab-475d-89ce-1bdaf85636cd	/user/data_silo1_load
rb-15730ae7-d3ba-40a1-924e-1cfbb143ba17	
rb-2d9514ed-f82c-4e24-a8a5-c74285e7f9dd	/user/data_silo1_load
rb-3fdbe962-1d6d-4139-8d06-00369791e1f5	/user/data_silo1_load
rb-52527b50-40fc-4220-983e-ebf2a3984422	
rb-5fee02a5-4c50-4e53-a91d-ad1e635959c0	/user/data_silo1_load
rb-6b5cbf40-7fbc-4a7c-8fec-3b78031e5518	
rb-93d450d0-e1ba-4b0e-ae53-2ff990f6f624	/user/data_silo1_load
rb-b3a7de45-f260-4fb3-88f3-b0a617862158	
rb-b9c48adf-2419-4cb1-af94-1add3a1491b2	/user/data_silo1_load
rb-bbca0256-b8ba-47cf-8227-a2bd089c184b	/user/data_silo1_load
rb-d0f069f7-e54c-49c4-afcc-5f2c8436cf9f	/user/data_silo1_load
rb-d66151bb-5e98-4c87-ad10-b6b65790715c	
rb-fcec7518-d760-47c8-9663-4dc2d67dfd80	
txt-1cf44712-0379-49bd-bd85-3f4fffc9ae78	/user/data_silo1_load
txt-84128e84-1e63-4d9b-b34d-d772f3997e1c	/user/data_silo1_load

Fig. 9. The metadata stored in HBase database

Fig. 10 shows that the redundant data in the DCM Cloud is logged in the Phoenix table. Data redundancy can be viewed and assessed by using SQL.

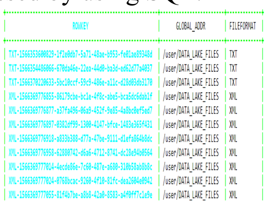


Fig. 10. Redundant data information in the table of Apache Phoenix

V. CONCLUSION

This paper presents a solution that uses selected big data software to address the unstructured data silo problem throughout the product life cycle. The problem is addressed through timely monitoring of the latest data; unstructured data integration and metadata management; data redundancy management; data inconsistency management with data provenance and data visualisation; and data control that applies pre-defined rules to automate workflow.

Although there is no big data software available to provide security for unstructured data on DCM Cloud, data control provides mechanisms to improve security on DCM Cloud. For example, if the USB interfaces of the DCM Edge computers are disabled and Internet access is blocked, the risk of data leakage can be reduced. Since DCM Cloud can be deployed in the manufacturers' intranet, the data of the upstream systems can still be transferred to the target systems through data control.

The proposed solution can be extended to integrate with more siloed data for analysis, optimization, prediction and support decision making. This work paves the way for the realization of various applications of Industry 4.0 and smart manufacturing such as digital thread, digital twin etc. The future research direction is to develop various applications that can be seamlessly embedded in the developed DCM architecture.

REFERENCES

[1] H. Henning, Kagermann. Wolfgang, Wahlster . Johannes, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Final Rep. Ind. 4.0 WG, no. April, p. 82, 2013.

[2] A. Kusiak, "Smart manufacturing must embrace big data," Nature, no. April, 2017.

[3] M. Helu, A. Joseph, and T. Hedberg, "A standards-based approach for linking as-planned to as-fabricated product data," CIRP Ann., vol. 67, no. 1, pp. 487–490, 2018.

[4] M. Helu, T. Hedberg, and A. Barnard Feeney, "Reference architecture to integrate heterogeneous manufacturing systems for the digital thread," CIRP J. Manuf. Sci. Technol., vol. 19, pp. 191–195, 2017.

[5] R. Lipman, "STEP File Analyzer Software," J. Res. Natl. Inst. Stand. Technol., vol. 122, no. 16, pp. 1–2, 2017.

[6] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," J. Manuf. Syst., no. January, 2018.

[7] Y. Cui, S. Kara, and K. C. Chan, "Manufacturing big data ecosystem: A systematic literature review," Robot. Comput. Integr. Manuf., vol. 62, no. January 2019, p. 101861, 2020.

[8] D. Vohra, "Apache Kafka," in Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools, Berkeley, CA: Apress, 2016, pp. 339–347.

[9] Y. Cui, S. Kara, and K. C. Chan, "Large Scale MTConnect Data Collection," in 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM) (CIS-RAM 2019), 209AD.