

## RESEARCH ARTICLE

# An Intelligent Product-Driven Manufacturing System Using Data Distribution Service

CHIN-TE LIN<sup>1</sup> AND HONG-JI LU<sup>1</sup>

Department of Mechanical Engineering, National Central University, Taoyuan 32001, Taiwan

Corresponding author: Chin-Te Lin (chintelin@ncu.edu.tw)

This work was supported by the Ministry of Science and Technology Council under Grant NSTC 112-2218-E-008-010.

**ABSTRACT** As key components of smart factories, intelligent production machines are gradually becoming a major capital asset of companies in the market, offering connectivity, interoperability, and services. However, due to commercial and technological constraints, production machines rarely provide enough information for customers to develop data-driven applications. Therefore, this research proposes that intelligence be built into them to assist in production control and allow relevant data to be collected and made available for future use. In order to realize this proposal, an operational paradigm for such systems is first outlined in this paper. Then, the states of workpieces in a production environment are classified, and the methods of generation, operation, and transfer of instances are described. The next step is to build an intelligent product-driven manufacturing system using Data Distribution Service (DDS). Intelligent workpieces and machines collaborate to complete production tasks in such a system. During this collaborative process, an intelligent workpiece continuously collects relevant data and packages it together for the customer on its transformation into a final product. The feasibility of this intelligent product-driven manufacturing system is examined by upgrading a classic manufacturing factory. In addition, the study explores the system architecture, the implementation of a classic system upgrade, the production data collection, and the intellectualization of data assets. In conclusion, the results of this study will help machine builders expand their business models, providing customers with production machines that carry data for advanced data-driven applications.

**INDEX TERMS** Data distribution service, intelligent product, manufacturing system, product life cycle, industry 4.0, industrial automation.

## I. INTRODUCTION

Production machines, such as CNC machining centers or industrial robotic arms, are an effective form of production asset by means of which goods are manufactured. Manufacturers of production machines add connectivity and intelligence to their products to assist customers (usually manufacturers of other goods) with condition monitoring, maintenance management, and process improvement to achieve goals of increased manufacturing efficiency or reduced costs. Although production machines often come with pre-defined functions and data, customers may modify the products or add peripherals to extend their applications.

The associate editor coordinating the review of this manuscript and approving it for publication was Okayay Kaynak<sup>1</sup>.

Also, customers may need more extensive product lifecycle data to develop data-driven intelligence features, such as prognostic and health management (PHM) [1] or digital twin [2], [3].

Product lifecycle can be segmented into the beginning of life (BOL) in the factory, middle of life (MOL) in the customer's application, and end of life (EOL) in the disposition process. Although the customers have enough detailed knowledge about MOL [4], accessing sufficient BOL data of production machines, like production approaches or quality control results, is not easy. This makes it difficult for customers to identify the differences between identically-marketed production machines, leading to confusion and inefficient use. For example, they do not know whether or not two machines of the same model require

different recipes to make the same goods and whether different maintenance plans must be used for them. Obviously, the interruption of data traceability between BOL and MOL restricts the degree of intelligence of production machines themselves or when integrated with other assets.

The difficulty of obtaining the production traceability of production machines is due to two reasons. First, the lack of mutual trust mechanisms can lead to machine manufacturers' unwillingness to share their manufacturing or usage data [5], [6]. Such commercial or legal concerns are research issues in supply chain integration and trust management and are beyond the scope of this study. The second reason is the lack of suitable ways for machine manufacturers to provide customers with access to relevant data. While manufacturers collect data that can help them improve product design and manufacturing process, it is also considered necessary to provide this data to customers in a way the latter can use for system planning, simulation, integration, and safety [7], [8], [9].

AutomationML, an XML-based format for handling plant engineering metadata, has been promoted to integrate engineering data formats [10]. Lüder et al. have demonstrated that the AutomationML format they developed based on process description does that and is a suitable way to provide the large amounts of information needed in the modeling of industrial processes [11]. AutomationML is also used to model mechatronic systems, improve automation, optimize production, and detect defects [12], [13], [14], [15]. AutomationML was applied to enrich ISA-95 to solve the lack of semantics of low-level components. Schmidt and Lüder summarized how to access or map data to AutomationML metamodels for use in the production life cycle [16]. While AutomationML supports consistent and lossless data exchange, the collection and storage of dynamic data in the manufacturing stage still lacks an implemented methodology.

Meanwhile, common sense dictates that cloud services be applied to manufacturing because of their many advantages, such as resource-sharing and openness. Hence, sharing data on the cloud is an available pathway that production traceability data sharing must support. For example, Helo et al.'s proposal of a cloud-based ecosystem to improve stakeholder collaboration takes such an approach [17]. However, as Jess et al. point out [6], all these approaches still face the stumbling block of companies' unwillingness to reveal their confidential information. Another issue is the storage costs of the machine manufacturer that maintains the collected data.

Therefore, an important motivation for this paper is to explore ways machine manufacturers can collect data for their purposes and provide the data to customers with suitable, economical methods. The authors believe that if machines in production are given intelligence at the manufacturing stage, they can collect data about themselves. In this way, customers can obtain relevant data directly from the machines after delivery without crossing the network or authenticating against external databases. To this end, this study proposes a new intelligent product-driven paradigm and develops a

manufacturing system based on the paradigm. The main goals of this study are to (1) Develop a paradigm for intelligent products to assist in data collection and manufacturing, including role, connectivity, functionality, and interactivity. (2) Design and apply Data Distribution Service (DDS) technology to implement a system realizing the intelligent product-driven manufacturing paradigm. (3) Adapt a classic manufacturing system in an experimental factory to conform to the proposed architecture and verify its feasibility. (4) Investigate the differences between intelligent product-driven and classic manufacturing systems.

The core contributions of this paper can be summarized in the following key points: (1) A unique distributed control architecture is proposed that leverages DDS technology and enables intelligent products to coordinate manufacturing operations with machines autonomously. (2) By delineating various process states, the authors establish the potential of intelligent products as dynamic data carriers while providing actionable insights for transitioning from centralized to distributed manufacturing systems. (3) An approach is introduced to intellectualizing product data, fostering adaptability and versatility within the manufacturing ecosystem. (4) Its analysis of system transmission tests offers a comprehensive examination of DDS network performance, emphasizing the influence of scalability and reliability on the proposed system.

The organization of this paper is as follows. Section II reviews the current status of the field and this study, discussing classic manufacturing systems and distributed manufacturing systems, the application of intelligent products in factories, the characteristics of middleware, the issues to be solved, and the study's objectives. Section III describes the study's paradigm for assistance with manufacturing, operations, and system integration using intelligent products. Included are the reasons for choosing DDS and the application design. Section IV reports the process and results of transforming the experimental classic manufacturing system into a distributed system. Section V summarizes the experimental results and the findings of the communication tests. Section VI discusses the differences between the distributed system and classic systems, the distributed system's advantages and disadvantages, and the features and limitations of the proposed network. Finally, Section VII summarizes the study.

## II. RELATED RESEARCH

### A. MANUFACTURING SYSTEM ARCHITECTURE

#### 1) CLASSIC MANUFACTURING SYSTEMS

The information system architecture of a real-world factory is usually based on the ANSI/ISA 95 standard [18]. In this architecture, components or modules are grouped into five hierarchical levels: machine, production line, floor, plant, and distribution, according to the objectives of the task, and then they are integrated between or within the hierarchical levels. A classic network architecture for manufacturing systems

consists of shop floor communication, business layer communication, and supply chain communication [19]. The need for connectivity and real-time requirements determines the components included in each communication network. A classic manufacturing system usually uses a centralized information computer center and a distributed control system (DCS). A DCS is usually a PLC-based bus network that provides good configurability, reliability, and ease of maintenance of the production line through inter-connected and conditional signals. Because of the simple architecture and ease of implementation, classic manufacturing systems are widely used in the real world. However, classic manufacturing systems often face several drawbacks: (1) The centralized message process under the control of the computer center creates single points of failure that can render the whole system inoperable. (2) The tightly coupled components make it difficult to expand the system. (3) Changing processes requires adjustments to the processing logic within the computer center, thereby creating potential system instability and risk.

## 2) DISTRIBUTED SYSTEM ARCHITECTURE

As information and communication technologies evolve, existing ANSI/ISA 95 standard-based hierarchical industrial information systems are expected to make the transformation to interactive architectures across hardware or functional levels through CPS [20]. Network and virtual object-based CPS can bring greater flexibility, efficiency, adaptability, and reliability to manufacturing systems. In addition, intelligent product and manufacturing system interactions can bring benefits such as increased product tracking accuracy, reduced inventory levels, and automated inventory management by providing real-time product data [21]. When every individual unit can communicate with each other to organize itself and work together, this distributed manufacturing system is also known as a Holonic manufacturing system [22], [23], [24], [25]. The differences and evolution of the various Holonic manufacturing systems are discussed in a review study by Kaiser et al. [26]. The main characteristics of Holonic manufacturing systems are scalable modularity, high flexibility and adaptability, reliable distributed decision-making, autonomy, and better reliability at the supervisory and planning levels in the automation pyramid. Because of these characteristics, distributed manufacturing systems are considered the future trend.

## B. INTELLIGENT PRODUCTS IN MANUFACTURING

McFarlane et al. introduced the concept of intelligent products, which they defined as a product that has a unique identifier, can effectively communicate with its environment, can hold its relevant data, can present its features or production requirements, and can participate in or determine its destiny. The concept has become a reality observed everywhere, driven by Industry 4.0 technologies such as the Internet of Things, big data, cloud services, and artificial intelligence. To clarify the definition of intelligent products, Raff et al.

selected 38 key papers from the Web of Science (WoS) database, analyzed the content, and, on the basis of prototypes and definitions, classified intelligent products into four different categories, namely, digital, connected, responsive, and intelligent. The authors identified the historical development of production machines as following their classification. Production machines were digitized early on in order to automate production. During this period, manufacturers targeted the manufacturing functions of their production machines to meet efficiency-oriented needs. From around the 2000s, production machines gradually started offering limited connectivity, such as serial port uploading of programs. Then, contributing to the quality of inter-asset communication, data exchange standards, such as MTConnect up to the latest universal machine tool interface standard (umati) [27], [28], [29], are proposed and define that production machines should present data in the appropriate semantics. Now, production machines can continue collecting relevant data after being sold to customers to provide autonomous or customized services [30], [31]. This approach has enabled manufacturers to gradually become service providers [32], [33].

Connectivity-enabled Industry 4.0 components (I4C) are essential for achieving CPS and distributed architectures [34]. By adding administration shells made of software and communication technologies to production machines or components that have limited connectivity, assets can be upgraded to the I4C level with the ability to connect, respond, and provide more intelligent services [35], [36]. Some studies have demonstrated that products based on asset administration shells can be more flexible in meeting system integration requirements [37], [38], [39]. The authors argue that Industry 4.0 componentization can be regarded as an implementation of intelligent products. When production machines and components are interconnected and interoperable, a distributed manufacturing system can be efficiently integrated into the information system.

Modeling I4C for intelligent and autonomous applications is a topic that has recently received significant research attention from a number of different perspectives. One approach proposed by Grangel-González et al. is to develop semantic information models for administration shells so objects can be easily integrated into existing standards [40]. Zhang et al. have studied smart product design frameworks and product life cycle interaction strategies in the context of their environment interaction model [41]. Fleischmann et al. report that a subject-oriented modeling language, Parallel Activity Specification Schema, is suitable for designing I4C, especially for implementations with OPC UA [42]. Another I4C design framework based on multiple agents with different tasks and optimization has been proposed by Veiga et al. [43], [44]. These modeling framework studies explore the design purposes and implementation of I4C.

It is worth mentioning that related material to be produced and the work-in-progress (here referred to as workpieces) are not necessarily considered part of a distributed system. The workpieces that studies have called ‘smart items’ [45]

may lack sufficient application support to exhibit intelligence in practice. For example, due to a lack of power supply or active response capability, the workpieces might require tracking identification tags (e.g., RFID technology) or add-on electric devices and updating the corresponding status records on the storage media. Although the workpieces may have a physical entity and corresponding digital record, they behave passively in the system by being moved, stored, processed, and measured. In short, they are not given the intelligence to determine their own destiny.

### C. MIDDLEWARE FOR INDUSTRIAL SYSTEMS

In order to build smart manufacturing systems in factories, middleware can effectively reduce the complexities of integration work [46]. Middleware acts as a communication abstraction layer in a distributed system, allowing for the interoperability of subsystems built on different hardware platforms, operating systems, or programming languages. Standard middleware commonly found in industrial systems includes MQTT, OPC UA, ROS, and DDS [47]. Following is some of the middleware relevant to distributed systems:

- Message Queuing Telemetry Transport (MQTT) is a lightweight communication standard for performance-constrained hardware and low-bandwidth networks [48]. Based on the publish/subscribe (Pub/Sub) mechanism, MQTT provides hierarchical structure-based messages and uses brokers to store and forward messages to clients within the network. In addition, MQTT provides the basic quality of service (QoS) for setting message republication behaviors. Although MQTT is the most popular communication standard on the Internet of Things, it still has shortcomings in security, disorder, and reliability [49].
- OPC Unified Architecture (OPC UA) is a service-oriented communication protocol designed to enable machines as data source servers to provide relevant data and services to customers [28]. An initiative of the European manufacturing industry, OPC UA supports overarching industry application semantics, such as the injection molding machine standard (Euromap 77/83) and *umati* [29]. OPC UA recently started supporting Pub/Sub, but the native OPC UA Pub/Sub mechanism does not support QoS policies. OPC UA also provides a discovery mechanism based on local discovery servers so that newly registered machines can be found by existing participants [50].
- Data Distribution Service (DDS) technology is also communication middleware based on Pub/Sub mechanisms, but it provides more QoS settings to achieve reliability, priorities, and persistent control of message distribution. DDS is often used in industrial systems [51]. For example, the second generation of the Robot Operating System (ROS 2.0) is based on DDS [52]. The efficiency of DDS comes from the design of the global data space (GDS), where

pre-defined topics are registered and via which data can be transferred between writers and readers. Another important feature of DDS is the decentralized discovery mechanism, which allows participants to join the DDS network freely and exchange data through the GDS. These DDS features enable loose coupling, interoperability, and fine control of industrial automation [53].

### D. GAPS AND OBJECTIVES

Intelligent product development is gradually building autonomy and intelligence to improve productivity and quality in various fields. Moreover, in order to evaluate the characteristics of production machines more precisely, customers increasingly expect to acquire BOL phase data about the machines they purchase. The authors observed that the trend in machine communication standards [27], [28], [29] also implicitly assumes more BOL stage information from machine manufacturers. Important BOL stage data can help to predict equipment health [54], evaluate product performance by use of a virtual system [55], improve production efficiency or product quality [8], and ensure system safety [9]. However, production machines rarely come with historical information about their production process performance. Raw data is lacking in particular, so the data based on which users can develop applications is limited to that at hand. Furthermore, even given a relationship of trust established between the machine manufacturer and the customer, economical and easy-to-maintain ways to collect and preserve large volumes of raw data are needed. To do this, workpieces with basic intelligence can be deployed on the manufacturing floor and assist in the production process, collecting information about themselves and making it available directly to customers as an objective to pursue in the future.

## III. SYSTEM DESIGN

### A. PARADIGM

The product-service systems (PSS) lifecycle model developed by Cavalcante et al. is applied to analyze the activities of production machines, with results listed in Table 1. The PSS model divides the life cycle into four periods: Idea & Design, Realization, Use, and End of Life. The first two periods are usually regarded as BOL, as the product enters production by the machine manufacturer. After delivery, production machines help to produce the customer's goods during the Use period, and disposition marks the End of Life period. Data collection activities over the lifecycle of the intelligent production machine can be summarized as follows. In the Idea & Design period, the production machine should collect data on requirements, design specifications and documents, and production plans about itself. Over most of the Realization period, when the machine is on the shop floor of the machine manufacturer, production traceability data should be collected, including, for example, used facilities, process logs, and testing/inspection results. Then, the delivered intelligent production machine can provide data to the customer,

TABLE 1. The proposed operation forms of iProducts in their lifecycle.

PSS lifecycle model [33]	Idea & Design	Realization	Use	End of Life
Owner	Machine manufacturer		Customer	
Production Activity	<ul style="list-style-type: none"> <li>• Documentation of requirements and specifications</li> <li>• Product design</li> <li>• Production planning</li> </ul>	<ul style="list-style-type: none"> <li>• Manufacturing of products</li> <li>• Test or inspection</li> <li>• Data collection</li> <li>• Shop floor control</li> <li>• Delivery to customer</li> </ul>	<ul style="list-style-type: none"> <li>• Monitoring</li> <li>• Self-service</li> </ul>	<ul style="list-style-type: none"> <li>• Recycling</li> </ul>
iProduct stage	<i>Fully digitalization</i>	<i>Digital Twins</i>	<i>Autonomous operation</i>	
iProduct Main Feature	<ul style="list-style-type: none"> <li>• Digital form only for iWP</li> <li>• Existing in a boarding host</li> <li>• Easy and cost-effective for revision</li> <li>• Interaction with design and planning tools</li> <li>• Data collected by digital instance</li> </ul>	<ul style="list-style-type: none"> <li>• iWP with a physical instance in the shop floor and a digital instance in the boarding host</li> <li>• Responsive digital instance interacting with iMachines</li> <li>• Data collected by digital instance</li> <li>• Synchronization between digital instances and passive physical instances</li> </ul>	<ul style="list-style-type: none"> <li>• Autonomous iWP constructed by transferring its digital instance to the corresponding physical entity</li> <li>• Converting iWP to iMachine before delivery</li> <li>• Self-service product interacting with other products</li> </ul>	

who can collect further usage data for advanced services. In the End of Life period, all this collected data may be returned to the machine manufacturer for re-design of the product and its next-generation successors.

The interaction paradigm during the Realization period is plotted in Figure 1. In a smart factory, intelligent products can be divided into two types. The first type is machines or components that perform production activities, hereafter referred to as “intelligent machines (iMachines).” They are already well-connected and can accept orders to carry out manufacturing tasks. The second type is items that are to be produced or are in production, such as materials and unfinished products. For ease of classification, the second type of intelligent product is referred to here by the term “intelligent workpiece (iWP).” Their instances may be digital or physical, and the behaviors may be passive or active. During this period, iWPs can communicate with iMachines and other intelligent assets in the manufacturing system. i.e., they can independently collect data, analyze and judge the current situation, and make corresponding decisions independently. Also, at the end, the iWP will officially take on product status, an iMachine with intelligence and relevant data.

To construct the above intelligent product, a digital twin (DT) implementation framework is necessary. The combination of the administration shell concept and the multi-level modeling approach [56] provides a suitable guideline. The cyberspace iWP, i.e., the digital instance, must have a data interface, a virtual model to support physical instance mapping and a knowledge model to make decisions, as shown

in Figure 2. These models can help the iWP to perform production tasks autonomously. When the iWP achieves iMachine status, these models and the information collected during production will support the customer’s DT application. In addition, the iWP can be separately conceptualized as a complete virtualization, digital twins, and an autonomous operation, depending on the stage of the relevant operation during production.

### 1) FULL DIGITALIZATION

When a manufacturer accepts an order from a customer, the production task manager generates a digital instance of an iWP with information on design and production, such as component data, a digitalized geometric model, a kinematic mechanism, and a projection plan. The design information helps model the iWP virtual space, which will be mapped to its physical instance at the next stage. The knowledge space model constructed by the production plan will guide the activities of the iWP in the workshop based on its status. As production has not yet started, there is no physical body for the iWP hosted in the workshop. In this stage, all digital instances of iWP waiting for production are managed by a computer, their boarding host. The virtual and knowledge models, i.e., design and production planning, can be easily and cost-effectively revised in this form.

### 2) DIGITAL TWINS

After production starts, the iWP begins to have a physical instance as raw material or semi-finished product. The

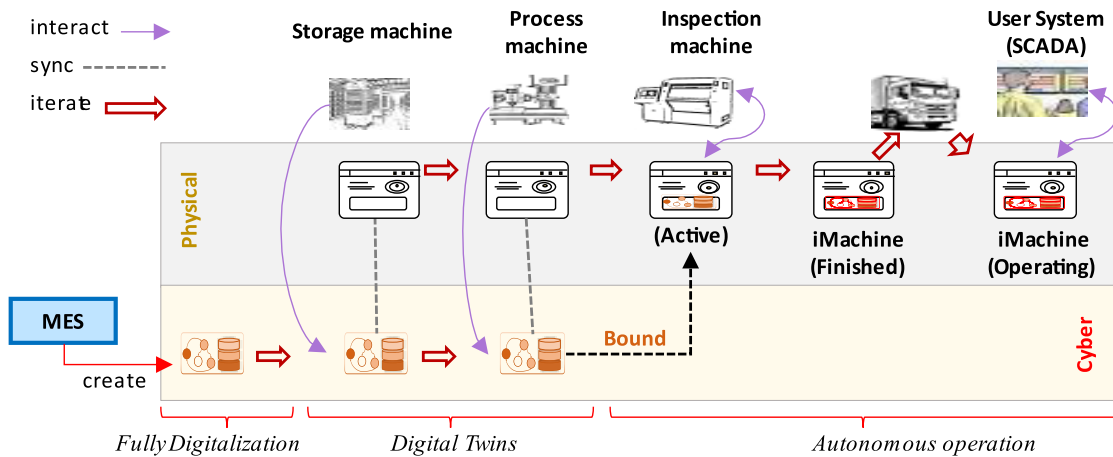


FIGURE 1. Intelligent production machine life cycle paradigm.

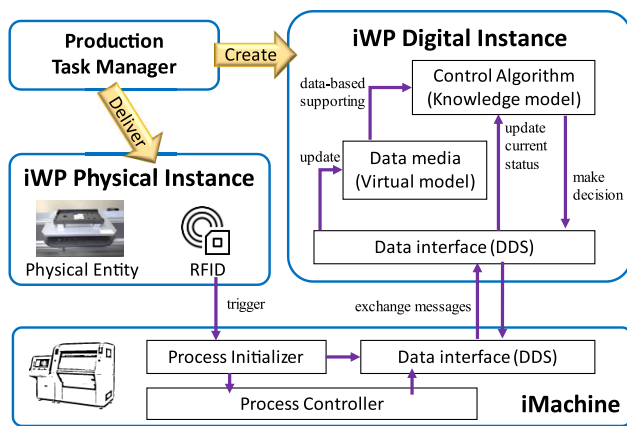


FIGURE 2. Instance states of the proposed manufacturing system.

physical instances may be temporarily not digitized due to the lack of corresponding computing hardware or the necessary instance power source. In this stage, iWPs exist on the production line as simple DT. i.e., they continue to be hosted on the original computer and are linked to the corresponding physical instances by its data interface and identification technology that synchronizes their statuses. The constructed knowledge space model in the iWP digital instance will direct its physical instance activities. Data from the manufacturing process is also synchronized with the iWP’s digital instance.

### 3) AUTONOMOUS OPERATION

With the merging of digital and physical instances, the iWP becomes autonomous in form and can communicate and interact directly with the manufacturing system. An autonomous iWP can interact directly with any intelligent product and perform production tasks. In this form, the iWP can directly collect data about its processes and decide its destiny.

Although the proposal is based on the analysis results of the Realization period, it has applicability to the other iWP periods. For example, the autonomous operations stage can be regarded as a kind of service to customers.

## B. INFORMATION SYSTEMS INTEGRATION

### 1) REASONS FOR CHOOSING DDS

Because the aim is to collect data from the iProduct relevant to production, the volume of messages of different sizes about the manufacturing system, including recipes, programs, documents, and images, can be expected to be significant. Also, compared to stationary iMachines, the status and connectivity of a dynamic iWP continuously change with progress in the production process, and this is expected to result in information management issues. For example, an iWP will probably switch among several local networks and collaborate with different objects during logistics. This situation requires overcoming the issues of participation and interaction. Therefore, choosing the most suitable communication middleware for this is the main concern in system integration.

The authors chose DDS for the following reasons. First, DDS has a clear advantage in its discovery mechanism, used to solve the issue of changing areas. When a workpiece enters a new area, intelligent products can quickly recognize each other, establishing their identities and enabling peer-to-peer connections. Second, DDS has a self-description feature, making it easier for intelligent products to communicate and collaborate. Finally, DDS supports a variety of QoS policies appropriate for different application scenarios and can properly handle different message priorities. For example, a wide variety of data for monitoring, controlling, and evaluating production status or product quality may be transmitted in the manufacturing system to meet production management requirements, with transmission latency times usually less than one second [57]. As a result, DDS is more flexible and configurable and thus more able to handle the various

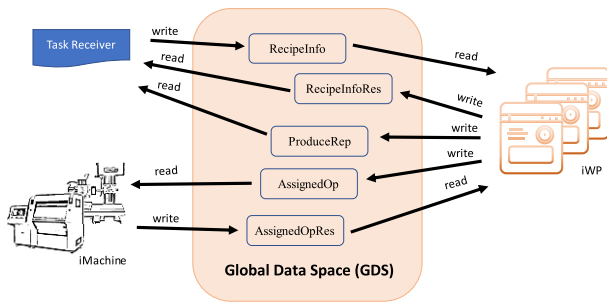


FIGURE 3. The GDS message map of the proposed manufacturing system.

issues involved in managing intelligent products than other middleware.

## 2) MESSAGES IN THE GDS

In the integration process for an industrial information system, it is necessary to clarify the requirements for message exchanges. This study requires a production plan for production when creating an iWP instance. When the iWP instance completes any task, it must record its production progress. In addition, iWP instances need to communicate with iMachines to perform operations and obtain process responses to determine their current production progress and status. Based on these requirements, the relevant topics of messages shown in Figure 3 are planned in the GDS of the DDS network. Initially, the task receiver will provide production recipe information to the iWP. The iWP will return a confirmed response on receiving the production recipe information. When the iWP is ready to engage with an iMachine, it assigns operation parameters and obtains process results upon operation completion. The iWP can also periodically report the production progress to the task receiver or finally. The iWP can collect data about itself, including its production plan and history, and direct the iMachine's actions through the above GDS design.

## 3) INSTANCE TRANSFER

The easiest way to move the digital instance of an iWP from its boarding host to the corresponding physical body is through data media. The content of iWP will be stored in the data media, such as XML, drawing, and SQLite files. With a well-designed interface combining data and control processes, instance transfer can be carried out by simply copying or moving the data media. Digital instance transfer involving functionality or interfaces may require a more complex approach, such as container technology. The behavior may be different before and after transferring. The concept is illustrated in Figure 4.

## IV. IMPLEMENTATION

### A. SYSTEM DEVELOPMENT

A proof of concept of the intelligent product-driven manufacturing system was carried out using the FESTO CP-Factory

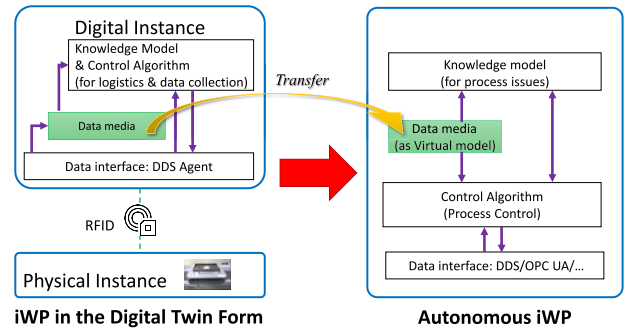


FIGURE 4. Illustration of a digital instance transfer.

shown in Figure 5(a). The hardware and software involved included an ASRS machine as an automatic storage system for materials and finished products, a MagBack machine for placing covers, an MPress machine as an assembly press, and a computer center on which the manufacturing execution system MES4 was installed. Conveyor belts and carriers achieve logistics between machines. The system's machines had embedded Siemens Simatic S7-1500 PLCs, which exchanged information in packets with a defined data structure in MES4 via standard TCP sockets [58]. Therefore, these machines can accept commands of the manufacturing execution system from the computer center to perform production tasks. The default FESTO CP-Factory with the architecture shown in Figure 5(b) can be considered a classic manufacturing system.

In order to implement the intelligent product-driven manufacturing system, the study had to re-make this classic manufacturing system. The steps were as follows:

- Step 1: A DDS network was built to enable the exchange of messages between the various physical nodes. The DDS used was eProsimas Fast DDS with version 2.0.0, a complete and free open-source DDS project based on the OMG DDS 1.4 and OMG RTPS 2.2 standards. The message implementation was coded in Interface Definition Language (IDL) to define the topics of Figure 2, allowing real-time message exchanges between physical nodes using the Pub/Sub mechanism. The DDS network comprised a cluster of nodes built with Raspberry Pi 4 Model B, containing a Broadcom BCM2711 quad-core Cortex-A72 processor and 4 GB LPDDR4 memory, and having the Ubuntu 20.04 LTS operating system based on the Linux kernel version 5.4.
- Step 2: To be a host for an iWP digital instance, the physical node hardware requires a pre-configured software environment. The environment included a DDS node process to connect to the GDS for data exchanges and an agent process to interact with other nodes. The iWP's digital instance, i.e., the SQLite content, determined its interactive behavior and the payloads of messages published to the other physical nodes.

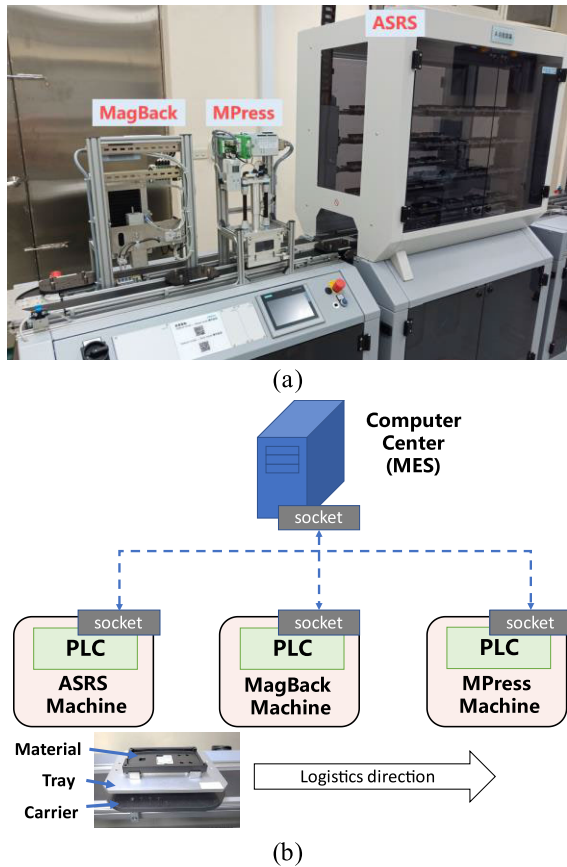


FIGURE 5. The classic manufacturing system: (a) Photograph; (b) Production flow.

- Step 3: Since the proposed distributed system did not have a computer center but only a management node, a Raspberry Pi could be used to handle the required production tasks, generate the digital iWP instances in the form of SQLite files, and transfer them to the physical nodes at the appropriate moments. Once the SQLite file of an iWP digital instance was transferred to its corresponding iWP hardware, the completed iWP could detect the environment and respond actively using the configured DDS and agent processes.
- Step 4: Connecting the machines to the DDS network was also important integration work. Message agents were established between the machines and the DDS network. The original PLCs of the machines already had OPC UA servers providing an interface to read machine information and control machine behavior, but they did not support DDS. Therefore, Raspberry Pi was deployed as an administration shell between the machines and the DDS network. With this, each machine and its administration shell was conceived as an iMachine, able to interact with other intelligent products through the DDS network.

As a result, the finished architecture is shown in Figure 6.

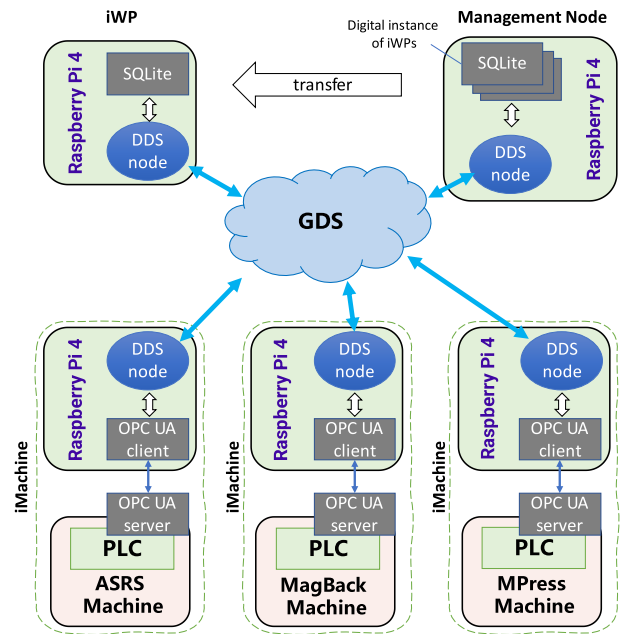


FIGURE 6. Intelligent product-driven manufacturing system architecture using DDS.

### B. DIGITAL INSTANCE OF INTELLIGENT WORKPIECE

In order to verify the feasibility of the system, an order of several products is sent to the already constructed system to carry out production. The iWP digital instance, which the management node generates, consists of data media and its administration shell (AS), as shown in Figure 7. The data media of the iWP digital instance uses SQLite to store the production plan, traceability data, and other information required by the system. The AS has a data interface to the DDS network, performs data exchanges, and analyzes data content. It has a listener that listens to the DDS network for information about the physical instance’s logistical status, start and stop times, and responses from the process. When the AS receives the instruction to start production, it will order the ASRS machine to bind the carrier, the physical material, and the digital instance through an RFID tag and activate the iWP’s digital twin state. Based on the data content and physical instance status, the AS can decide the action of the iWP, send process commands to the machine, and update production history and other data content. Please refer to Algorithm 1 for data agent control logic. Based on this logic, when a carrier enters any physical machine, its AS will determine whether to instruct the machine to perform an operation or exit the machine, ensuring accurate and consistent data collection. This process continues until the iWP has completed all the tasks of the production plan.

### V. PERFORMANCE EVALUATION

Relevant data collected from manufacturing systems may include recipe and status values, files, images, and video streams, and the payload sizes of this data will affect



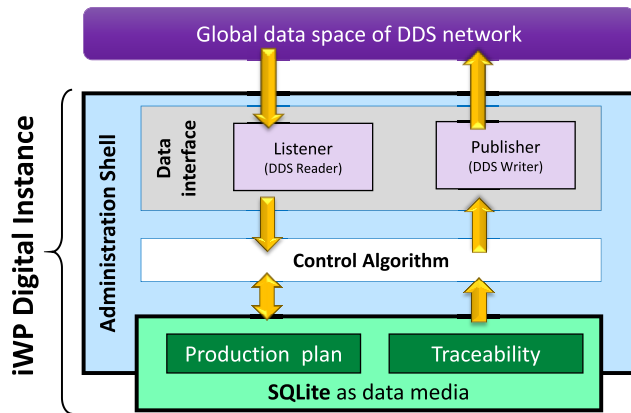


FIGURE 7. Illustration of iWP digital instance.

### Algorithm 1 Data Agent for Intelligent Workpiece (iWP)

```

Agent:
Create a SQL database for the iWP
Start Listener of DDS network
Initialize flag for Digital Twin

while true:
  if isDigitalTwin is false:
    if GUID in RecipeInfo message
      matches iWP's GUID:
      Commit production plan to database
      Set first step to nextStep
      Set Digital Twin flag to true
      Broadcast RecipeInfoRes message

  else isDigitalTwin is true:
    if nextStep does not exist:
      Commit Finish state to the database
      Broadcast ProduceRep message

  if a carrier arrives at a machine:
    if the carrier is empty at the ASRS:
      Set iWP bindedCarrierId to carrier
      Id

    if active machine Id matches that in
      nextStep:
      Broadcast assignOp message to start
      operation

  else:
    Broadcast assignOp message to
    release carrier

Listener:
Collect physical instance state
if production state is changed:
  Commit production state to the
  database
  
```

network transmission performance. In order to understand the actual benefits that a DDS network can provide intelligent product-based manufacturing systems or the limitations it may introduce, this section evaluates a series of system properties that include Round-Trip Times (RTT), message

reception rate, and throughput, which represent message latency, the number of messages, and the amount of data that can be transmitted in a given time unit, respectively. Almost all the QoS policies of the test used the default values provided by Fast DDS. Only the reliability setting was changed to “Reliable” in order to simulate the basic settings of a mission-critical message transmission. The hardware used for the performance test included four Raspberry Pi 4 Model B and a Dray VigorSwitch G1080 network switch, which provides a 1000 Mbps network interface, 16 Gbps switching capability, and network storm control for broadcast, multicast, and unicast. All Raspberry Pi were connected to the switch through RJ45 network cables.

### A. LATENCY

Firstly, RTT message transmission tests were carried out to see whether DDS moved messages between nodes quickly, so it was an effective production management assistance tool. Production management processes often create messages in the form of queries and responses that round-trip back and forth between nodes, like the pair *AssignedOp* and *AssignedOpRes* in Figure 2. A production order is only confirmed after the sender has received a response from the receiver. This means that an RTT value is a measure of the timeliness of the DDS network in carrying out such production management tasks. In the paradigm followed by this study, most of the iWP's messages in the fully digitalized or digital twin stage were transmitted within a single host. After taking physical form, the iWPs transferred messages across hosts. Therefore, this section describes *in-host tests* within the same host and *cross-host tests* between hosts on different Raspberry Pi. The tests were conducted in unicast mode, using different payloads, starting from 8 bytes and increasing to 32 KiB by an exponent of 2. RTT values were recorded 1000 times, and their characteristics were analyzed.

A breakdown of the results for the in- and cross-host RTT tests over the different payload lengths is presented first. Figure 8 shows the statistical distribution of in-host RTT. As the payload length increased, there was a gradual RTT increase, but it was negligible, doubling for a 2048-fold payload length increase from 8 bytes to 32 KiB. It should be added that a percentage (less than 8%) of anomalous RTT values in the in-host tests exceeded 1000  $\mu$ s, but they are not plotted in Figure 8. We suggest that the delay for the larger payloads is due to the performance limitations of the Raspberry Pi 4B hardware.

Figure 9 shows the statistical distribution of the cross-host RTT tests. Compared with the average in-host test values of 200 to 300  $\mu$ s, cross-host RTT values were significantly higher in the 200 to 5000  $\mu$ s range. Significant jitter, i.e., variation in RTT values, can be observed in Figure 9. For example, in Figure 9(a), two groups of message transmission times can be observed, one at 600  $\mu$ s and the other at 2900  $\mu$ s. As the payload length increased, jitter and latency increased gradually, as seen in Fig. 9(b).

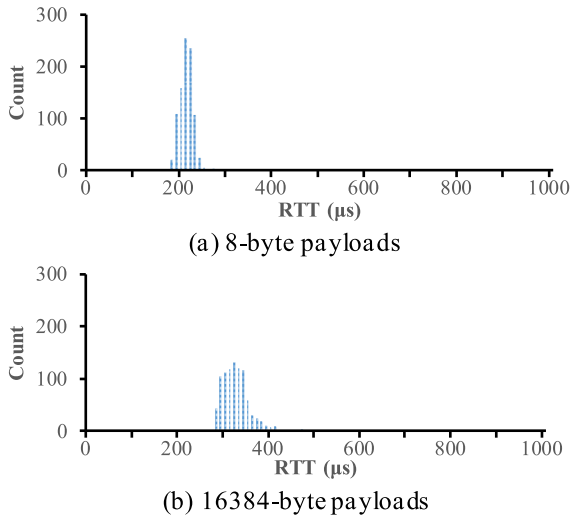


FIGURE 8. In-host testing RTT distribution.

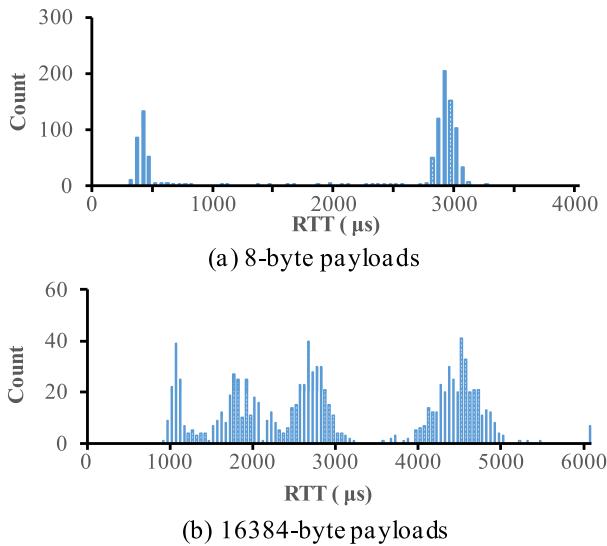


FIGURE 9. Cross-host testing RTT distribution.

Standard statistical metrics are given in Table 2, and average RTT values are plotted in Figure 10, providing a general overview of RTT performance. In-host transmission performance was undoubtedly very good, with high speeds and low variation. Most messages were transmitted within hundreds of microseconds. Cross-host transmission performance was mainly affected by the performance of the switch. There was performance degradation of 7.5 times, approaching an order of magnitude, over in-host times, accompanied by greater variation. When the payload size exceeded 1 KiB, the variation in RTT started to increase. At sizes greater than 4 KiB, transmission performance degradation increased significantly due to limitations imposed by the network’s maximum transmission unit (MTU), usually 1,500 bytes per packet in an Ethernet network. When the length of the packet enclosing the payload was larger than the MTU, the

TABLE 2. Round-trip time tests statistic results.

Payload (Bytes)	in-host			cross-hosts		
	Min (μs)	Avg (μs)	SD (μs)	Min. (μs)	Avg (μs)	SD (μs)
8	173	207.3	19.4	276	2891.9	70.2
16	171	205.3	18.5	266	2994.8	79.5
32	173	208.0	29.8	270	2918.9	78.3
64	174	208.3	39.6	274	2967.5	75.4
128	177	210.0	20.7	281	2916.8	72.2
256	180	221.8	31.6	294	3049.0	90.1
512	182	222.2	33.5	299	2997.8	76.4
1024	267	322.2	38.3	319	3121.9	85.1
2048	186	225.0	35.7	425	3292.9	202.9
4096	189	225.4	24.3	479	3313.8	218.0
8192	402	484.7	39.8	622	3636.7	333.1
16384	200	238.4	23.5	883	4431.0	269.6
32768	223	266.5	26.3	1540	4386.2	368.2

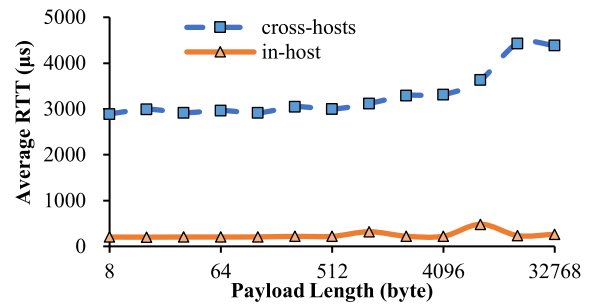


FIGURE 10. Average RTT of in-host and cross-host over various payload lengths.

transmission was divided into several packets, and a delay resulted. Nevertheless, the results show that DDS always met real-time production management requirements, i.e., production dispatch or SCADA usually requires only one-second transmission performance.

**B. MESSAGE RECEPTION RATE AND THROUGHPUT**

In order to further understand the impact of message length on the collection of iProducts receiving information and reporting on states in the factory, the throughput of the DDS network with payloads of different lengths was then tested. In each test, the publisher continuously sent subscribers as many messages as possible of the specified length. Each test recorded the number of messages subscribers received in 100 seconds to measure message reception rate (MRR) and throughput. All tests were conducted in cross-host mode.

**1) MULTICAST TEST**

Multicast performance was tested to simulate the process of a task receiver distributing work to iWP. In the test, one publisher continuously delivered messages of a given length to multiple subscribers. The test variables were the number of assigned subscribers, ranging from 1 to 3, and the payload length, ranging from 8 bytes to 8 MiB.

The test results are shown in Fig. 11. Changes in MRR for a given number of subscribers were insignificant when

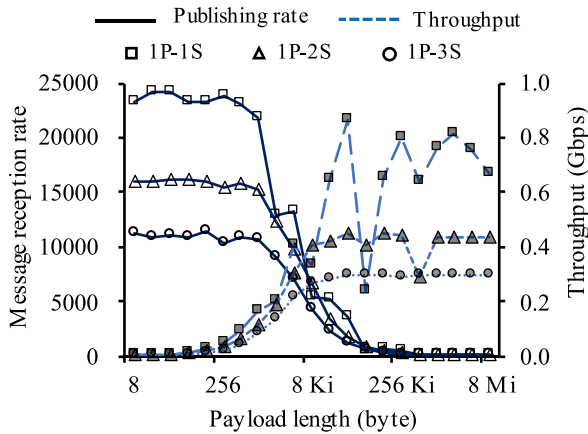


FIGURE 11. The relationships among payload length, message reception rate, and throughput in multicasting.

the message length was less than the network MTU. In this case, MRR can be regarded as a constant. When the message length exceeded the MTU, MRR decreased gradually. The relationship between it and throughput approached inverse proportionality. As MRR decreased with data length increases, throughput continued to increase until it reached saturation; after that, it can also be regarded as a constant.

Based on the above observations, a generic evaluation of the MRR can be expressed in the following form,  $P_{N,L}$

$$P_{N,L} = \begin{cases} \frac{P_{1,MTU}}{N} & \text{if } N = 1 \& L < MTU \\ \frac{P_{1,MTU} \cdot MTU}{N \cdot L} & \text{otherwise} \end{cases} \quad (1)$$

where  $N$  is the number of subscribers, and  $L$  is the message length. For the Raspberry Pi used in this study, the  $P_{1,MTU}$  obtained after fitting is about 24550. A comparison between this fitting result and the actual test values is shown in Figure 12.

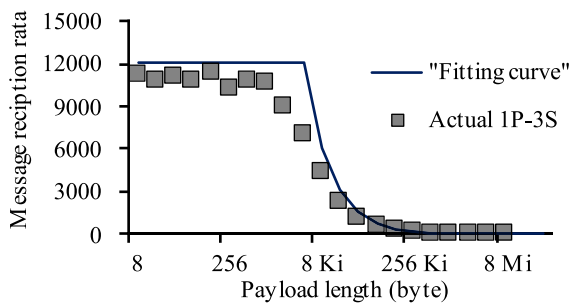


FIGURE 12. The evaluated message reception rate of various payload lengths.

Table 3 shows theoretical evaluation results for larger numbers of subscribers obtained from Equation (1). When the message length is less than MTU, for 10000 subscribers, the network can provide an MRR of about 2.46, which is sufficient for factories where recipes only require process parameters. For larger payload recipes and hundreds of

TABLE 3. Theoretical message reception rates with multicasting.

subscribers Payload Length	1	10	100	1000	10000
< MTU	24550	2455	245.5	24.55	2.46
4 KiB	8990	899	89.9	8.99	0.90
64 KiB	561.9	56.2	5.6	0.56	0.06
1 MiB	35.1	3.51	0.35	0.04	0.004
8 MiB	4.4	0.4	0.04	0.004	0.0004

machines, the system will likely take longer to send messages, and it is hard to expect it to be successful in most such cases. However, since actual recipe assignments are infrequent and content length is often less than the MTU, the DDS network can still satisfy the requirements of management tasks requiring multicast in many cases.

## 2) REPORTING TEST

In contrast to the case of one-to-many multicasting, reporting is done by multiple iProducts to a few or only one management system to update it/them on the status of production progress. In this case, reporting performance is a function of the limits of the management system’s ability to handle large messages. In a reporting test, multiple publishers continuously send messages with a given length to a specified subscriber. Test variables are the number of publishers, ranging from 1 to 3, and the payload length of the message, ranging from 8 bytes to 8 MiB.

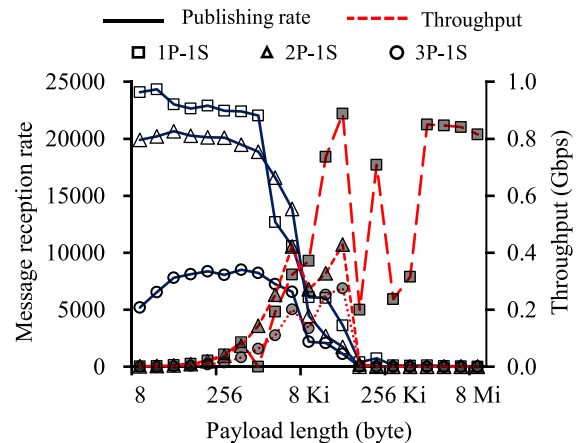


FIGURE 13. The relationships among payload length, message reception rate, and throughput in reporting.

Figure 13 shows the results of the reporting performance tests. It can be seen that the main pattern is similar to that of the multicast performance tests, i.e., MRR decreases, and throughput reaches a limit as both payload length and the number of iProducts involved increase. However, while MRR behavior remained the same as in the previous tests, in the manner explained by Equation (1), throughput, with message reception count, reached zero for payload lengths that

exceeded 64 KiB, i.e., the subscriber did not receive any messages at all, when receiving messages from 2 or more publishers. This means that DDS transmission is unsuitable for large payloads near or over 64 KiB in length. Reducing the size of data or splitting it into multiple messages may be the only way large amounts of data can be transmitted.

## VI. DISCUSSION

The aim of this research was to develop an intelligent product-driven manufacturing system that allows customers to write data-driven applications using relevant data that their products autonomously collect. The following describes the results of this research and other studies:

1. **Distributed Control Architecture:** A classic manufacturing system operates with a computer center directing the interactions between the machines and the workpieces. In this study, control and tracking of the intelligent product were distributed so that the machines and intelligent products could directly coordinate and conduct manufacturing operations themselves. To this end, the necessary processes and forms of intelligent products were developed, and system architecture was designed using DDS technology and its particular message patterns. Finally, prototype tests in an experimental plant verified that the intelligent products, as designed, could direct the manufacturing process and collect relevant data.
2. **State Design of Intelligent Products:** The study described intelligent products and their states as fully digital, digital twins, or autonomous. Each form's operations and transfer mechanisms were implemented in the experimental factory. Therefore, the study established the feasibility of using intelligent products as data carriers. However, focusing on the DT form, i.e., when all the digital instances of products exist on a single computing device, their operational behavior is similar to that of a classic manufacturing system. The main difference between the two is whether the digital instances can exist independently of their boarding host or not. In view of this relationship, the approach proposed here can serve as a reference if one wants to transform a centralized manufacturing system into a distributed manufacturing system.
3. **Data Collection and Transfer:** The instances in the system collected relevant information on mounted data media when exchanging information through the DDS network. This economical approach meant information related to production control and manufacturing was distributed only to entities that needed it, avoiding data exposure, a risk with network transmission, and reducing the cost of data storage compared to storing this information in the cloud or on shared media. This mechanism also provides machine manufacturers with a more economical and secure way to store BOL data, increasing the benefit of data transfer to customers with their products.
4. **Administration Shell for Data:** During implementation, it was observed that the essence of the iWP digital instance is data media. In order to allow data to interact with other intelligent products, the approach of this study added an administration shell to data media. A low intelligence was added to the agent based on the multi-level modeling approach [56] for the development of digital twins. This approach is similar to that of several AutomationML studies [11], [16], [59]. The difference with those studies is that this one suggests that the administration shells for combining data media should also adapt to the situation. For example, the iWPs in this study use management nodes and workpiece entities as administration shells for different usage purposes. The contribution of this study is the intellectualization of the data in the BOL stage of intelligent products.
5. **Latency and Jitter:** As the amount of data transferred across hosts increased, there was a gradual increase in both latency and jitter, indicating that the system was under greater processing load, degrading transmission performance. Secondly, when packet size exceeded the maximum transmission unit (MTU), the packet was split into several smaller packets for transmission, thereby causing critical additional processing time. These results may serve to remind users of the need to ensure that packet size does not exceed the network's MTU when designing message structures to avoid unnecessary delays. Despite these observed variations in transmission performance, the experimental results show that the DDS system can provide stable and reliable transmission performance under known conditions, ensuring the smooth operation of production management.
6. **Multicast Performance:** Several key observations can be made about the multicast tests in which a publisher transmits messages to multiple subscribers. When the message length was below the MTU, MRR took generally unvarying, constant values inversely dependent on the number of subscribers. Exceeding the MTU, MRR decreased gradually, and the relationship between it and throughput was inversely proportional. As data length increased further, throughput increased until saturation. On the basis of this use of Fast DDS and Raspberry Pi to collect performance data and establish simple estimation methods, we have come to several conclusions in evaluating the limitations and applicability of various schema. For task dispatches that require only process parameters, a DDS network can reliably serve 10,000 iProducts. However, for complex and larger payloads, such as complex G-code programs for CNC machining centers that often involve the transfer of tens of KiB, performance will drop to less than one transfer per second serving hundreds of machines. Payload lengths greater than 64 KiB create challenges for message distribution, for example, in Augmented

Reality or Virtual Reality applications. Still, the DDS network built into the Raspberry Pi will satisfy most real-world management needs despite the relatively low distribution frequency of the actual processor.

7. Report Performance: When multiple publishers sent messages to the same target, performance was similar to that of multicasting, i.e., MRR and throughput remained constant or decreased with the increase of payload length and the number of publishers. Beyond data lengths of about 64 KiB, however, both dropped to zero, meaning the subscriber did not receive any messages at all. Data that is too large may not be suitable for DDS transmission, and it is recommended that alternative methods be used, such as reducing the length of the data and splitting it into multiple publications.

## VII. CONCLUSION

This study proposed a distributed manufacturing system framework using intelligent products as its main production control unit and DDS as its communication middleware and validated it in an experimental factory. In this framework, intelligent products acted as production control units in the production process and submitted the data they collected to the customer. This approach gives data a status approaching that of intelligent products. At the same time, the performance of DDS as middleware was tested and evaluated in the factory. The results show that although transmission in the DDS network was affected by delay and jitter, production management tasks were not affected. MRR and throughput depended on the number of multicast targets and message length, but this was not a problem for small messages of length less than the MTU. However, their impact must be considered when transferring larger amounts of data. In summary, the study lays out a new intelligent product-driven manufacturing system approach that will contribute to developing innovative production management methods and business models for manufacturers.

Compared to real-world industrial manufacturing systems, the manufacturing system designed, implemented, and analyzed in this study is simple, using an experimental factory and focusing on single-assembly production control. In addition, the investigation of digital asset management and control at the BOL stage was limited. Therefore, the study's system integration work and discussion may not apply to real-world factories. Because of these limitations, possible future research directions include:

1. Expansion of the scope of system integration would provide more comprehensive product-related information relevant to a wider range of applications and services targeting, for example, the supply chain and product life cycle. This expansion would allow the collection of product-related information more effectively and provide more information for customers optimizing their data-driven applications and services. At the same time, if the proposed system framework were to

be applied to a real-world factory, developers would have to be aware of the changes to processes that might result from the distributed system and the impact on system architecture. In addition, protecting the data in smart products and disclosing them only to customers is also an important data privacy issue.

2. It is worth exploring the value that can be derived from data by studying administration shells for different data types and sources and exploring their interactions. Specific examples include computer-aided design (CAD) data to help customers build virtual factories or computer-aided engineering (CAE) data to develop digital twins.
3. Constructing a system complexity analysis methodology and applying it to system improvement is a third research avenue. If the complexity of a system can be quantified, then processes, architectures, and message routes can be adjusted and optimized based on metrics. Such improvements would enhance a system's performance, flexibility, and scalability.
4. The manufacturing system considered in this study was primarily one of flow production, where physical tracking and transfer of the iWP were simple. In the case of a project-based production system, for example, where the production process involves multiple assemblies, the paradigm of this study would need further elaboration, and the work incorporating the information system would need to be carefully carried out. Even so, the operational forms and instance transfers presented in this study provide the basis for building advanced approaches to complex production situations.
5. Once intelligent products can determine their destiny, it is expected that they will be able to assist in many more production activities, including smart logistics, quality improvement, and energy optimization. For such purposes, more advanced middleware or reference architectures should be considered. For example, a multi-edge access framework [60] may be suitable for developing applications based on the wisdom of crowds and intelligent products.

For subsequent research, researchers should also use ROS2 and container technology. DDS requires greater software development skills and currently lacks adequate community support compared to other popular communication mechanisms. ROS 2 technology, based on DDS, benefits from many online resources, allowing for accelerated exploration of system planning and research into more complex manufacturing behavior.

## REFERENCES

- [1] J. Zendoia, U. Woy, N. Ridgway, T. Pajula, G. Unamuno, A. Olaizola, A. Fysikopoulos, and R. Krain, "A specific method for the life cycle inventory of machine tools and its demonstration with two manufacturing case studies," *J. Cleaner Prod.*, vol. 78, pp. 139–151, Sep. 2014.

- [2] N. Yousefnezhad, A. Malhi, T. Kinnunen, M. Huotari, and K. Främling, "Product lifecycle information management with digital twin: A case study," presented at the IEEE 18th Int. Conf. Ind. Informat. (INDIN), Warwick, U.K., Jul. 2020.
- [3] D. Lehner, S. Sint, M. Vierhauser, W. Narzt, and M. Wimmer, "AML4DT: A model-driven framework for developing and maintaining digital twins with AutomationML," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2021, pp. 1–8.
- [4] J.-H. Shin, D. Kiritsis, and P. Xirouchakis, "Design modification supporting method based on product usage data in closed-loop PLM," *Int. J. Comput. Integr. Manuf.*, vol. 28, no. 6, pp. 551–568, Jun. 2015.
- [5] M. Zhang and B. Huo, "The impact of dependence and trust on supply chain integration," *Int. J. Phys. Distribution Logistics Manage.*, vol. 43, no. 7, pp. 544–563, Aug. 2013.
- [6] T. Jess, P. Woodall, and D. McFarlane, "Overcoming limited dataset availability when working with industrial organisations," in *Proc. IEEE 13th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2015, pp. 826–831.
- [7] S. Faltinski, O. Niggemann, N. Moriz, and A. Mankowski, "AutomationML: From data exchange to system planning and simulation," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2012, pp. 378–383.
- [8] K.-P. Lin, C.-M. Yu, and K.-S. Chen, "Production data analysis system using novel process capability indices-based circular economy," *Ind. Manage. Data Syst.*, vol. 119, no. 8, pp. 1655–1668, Sep. 2019.
- [9] Y. Keshun, Q. Guangqi, and G. Yingkui, "Remaining useful life prediction of lithium-ion batteries using EM-PF-SSA-SVR with gamma stochastic process," *Meas. Sci. Technol.*, vol. 35, no. 1, Jan. 2024, Art. no. 015015.
- [10] A. Lüder and N. Schmidt, "AutomationML in a nutshell," in *Handbuch Industrie 4.0: Produktion, Automatisierung Und Logistik*, B. Vogel-Heuser, T. Bauernhansl, and M. ten Hompel, Eds. Berlin, Germany: Springer, 2016, pp. 1–46.
- [11] A. Lüder, L. Hundt, and A. Keibel, "Description of manufacturing processes using AutomationML," in *Proc. IEEE 15th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2010, pp. 1–8.
- [12] M. Vathoopan, H. Walzel, W. Eisenmenger, A. Zoitl, and B. Brandenbourger, "AutomationML mechatronic models as enabler of automation systems engineering: Use-case and evaluation," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, Sep. 2018, pp. 51–58.
- [13] A. Mazak, A. Lüder, S. Wolny, M. Wimmer, D. Winkler, K. Kirchheim, R. Rosendahl, H. Bayanifar, and S. Biffl, "Model-based generation of run-time data collection systems exploiting AutomationML," *Automatisierungstechnik*, vol. 66, no. 10, pp. 819–833, Oct. 2018.
- [14] P. Novák, F. J. Ekaputra, and S. Biffl, "Generation of simulation models in MATLAB-simulink based on AutomationML plant description," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7613–7620, Jul. 2017.
- [15] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with AutomationML and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.
- [16] N. Schmidt and A. Lueder, "The flow and reuse of data: Capabilities of AutomationML in the production system life cycle," *IEEE Ind. Electron. Mag.*, vol. 12, no. 2, pp. 59–63, Jun. 2018.
- [17] P. Helo, Y. Hao, R. Toshev, and V. Boldosova, "Cloud manufacturing ecosystem analysis and design," *Robot. Comput.-Integr. Manuf.*, vol. 67, Feb. 2021, Art. no. 102050.
- [18] *Enterprise-Control System Integration—Part 1: Models and Terminology*, Standard IEC 62264-1:2013, Int. Electrotech. Commission, Geneva, Switzerland, 2013.
- [19] T. O. Boucher and A. Yalçin, "Introduction," in *Design of Industrial Information Systems*, vol. 1. New York, NY, USA: Academic, 2006, pp. 1–25.
- [20] B. Vogel-Heuser, G. Kegel, and K. Wucherer, "Global information architecture for industrial automation," *Atp-Automatisierungstechnik*, vol. 51, nos. 1–2, p. 108, Jun. 2013.
- [21] D. McFarlane, S. Sarma, J. L. Chirn, C. Y. Wong, and K. Ashton, "The intelligent product in manufacturing control and management," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 49–54, 2002.
- [22] P. Valckenaers, H. Van Brussel, L. Bongaerts, and J. Wyns, "Holonc manufacturing systems," *Integr. Comput. Aided Eng.*, vol. 4, no. 3, pp. 191–201, Jul. 1997, doi: 10.3233/ICA-1997-4304.
- [23] J.-L. Chirn and D. McFarlane, "Evaluating holonic control systems: A case study," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 211–216, 2005.
- [24] D. C. McFarlane and S. Bussmann, "Developments in holonic production planning and control," *Prod. Planning Control*, vol. 11, no. 6, pp. 522–536, Jan. 2000.
- [25] C. Indriago, O. Cardin, N. Rakoto, P. Castagna, and E. Chacòn, "H<sup>2</sup>CM: A holonic architecture for flexible hybrid control systems," *Comput. Ind.*, vol. 77, pp. 15–28, Apr. 2016.
- [26] J. Kaiser, D. McFarlane, G. Hawkrigde, P. André, and P. Leitão, "A review of reference architectures for digital manufacturing: Classification, applicability and open issues," *Comput. Ind.*, vol. 149, Aug. 2023, Art. no. 103923.
- [27] (Aug. 2023). *MTCconnect Standard—Part 1. Fundamentals, Version 2.2.0*. [Online]. Available: <http://www.mtconnect.org/>
- [28] (Nov. 2017). *OPC Unified Architecture: Specification—Part 1: Overview and Concepts, Version 1.0.4*. [Online]. Available: <https://opcfoundation.org/>
- [29] (Jul. 2022). *OPC UA for Machine Tools—Part 1: Machine Monitoring and Job Overview, Version 1.01.1*. [Online]. Available: <https://reference.opcfoundation.org/MachineTool/v100/docs/>
- [30] X. Yang, P. Moore, and S. K. Chong, "Intelligent products: From lifecycle data acquisition to enabling product-related services," *Comput. Ind.*, vol. 60, no. 3, pp. 184–194, Apr. 2009.
- [31] Amazon Web Services. (2024). *Manufacturing on AWS*. Accessed: Jan. 22, 2024. [Online]. Available: <https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/manufacturing-on-aws-ra.pdf>
- [32] M. Schroeck, A. Kwan, and J. Kawamura, "Ecosystem-driven portfolio strategy: Building a portfolio of digital industrial solutions leveraging powerful business ecosystems," in *Digital Industrial Transformation*. New York, NY, USA: Deloitte Insights, 2019. [Online]. Available: <https://www2.deloitte.com/xe/en/insights/focus/industry-4-0/business-ecosystem-strategy.html>
- [33] J. Cavalcante and L. Gzara, "Product-service systems lifecycle models: Literature review and new proposition," *Proc. CIRP*, vol. 73, pp. 32–38, Jul. 2018. [Online]. Available: <https://doi.org/10.1016/j.procir.2018.03.324>
- [34] M. Hoffmeister, "Industrie 4.0: The Industrie 4.0 component," *Plattform Industrie 4.0*, Apr. 2015. [Online]. Available: <http://85.214.218.90/platform/content/documents/ZVEI-Industrie-40-Component-English.pdf>
- [35] C. Diedrich, A. Belyaev, T. Schroder, J. Vialkowsch, A. Willmann, T. Usländer, H. Kozirolek, J. Wende, F. Pethig, and O. Niggemann, "Semantic interoperability for asset communication within smart factories," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.
- [36] H. Bedenbender et al., "Relationships between I4.0 components—Composite components and smart production," *Plattform Industrie 4.0*, Feb. 2018. [Online]. Available: <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/relationships-i40-components.html>
- [37] Z. Bradac, P. Marcon, F. Zezulka, J. Arm, and T. Benesl, "Digital twin and AAS in the Industry 4.0 framework," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 618, no. 1, Oct. 2019, Art. no. 012001.
- [38] L. Neto, P. Torres, R. Dionisio, S. Malhão, and G. Gonçalves, "An Industry 4.0 self description information model for software components contained in the administration shell," presented at the 8th Int. Conf. Intell. Syst. Appl., Jul. 2019.
- [39] X. Ye and S. H. Hong, "Toward Industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019.
- [40] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana, and M. Hoffmeister, "Towards a semantic administrative shell for Industry 4.0 components," in *Proc. IEEE 10th Int. Conf. Semantic Comput. (ICSC)*, Feb. 2016, pp. 230–237.
- [41] H. Zhang, S. Qin, R. Li, Y. Zou, and G. Ding, "Environment interaction model-driven smart products through-life design framework," *Int. J. Comput. Integr. Manuf.*, vol. 33, no. 4, pp. 360–376, Apr. 2020.
- [42] A. Fleischmann, A. Friedl, D. Großmann, and W. Schmidt, "Modeling and implementing of Industrie 4.0 scenarios," in *Modelling To Program*, A. Dahanayake, O. Pastor, and B. Thalheim, Eds. Cham, Switzerland: Springer, 2021, pp. 90–112.
- [43] J. T. Veiga, M. A. O. Pessoa, F.-C. Junqueira, P. E. Miyagi, and D. J. dos Santos Filho, "Control of manufacturing systems by HMS/EPS paradigms orchestrating I4.0 components based on capabilities," in *Technological Innovation for Applied AI Systems*, L. M. Camarinha-Matos, P. Ferreira, and G. Brito, Eds. Cham, Switzerland: Springer, 2021, pp. 62–70.

- [44] J. T. Veiga, M. A. O. Pessoa, F. Junqueira, P. E. Miyagi, and D. J. dos Santos Filho, "A systematic modelling procedure to design agent-oriented control to coalition of capabilities—In the context of I4.0 as virtual assets (AAS)," *Computers*, vol. 10, no. 12, p. 161, Nov. 2021.
- [45] D. McFarlane, V. Giannikas, A. C. Y. Wong, and M. Harrison, "Product intelligence in industrial control: Theory and practice," *Annu. Rev. Control*, vol. 37, no. 1, pp. 69–88, Apr. 2013.
- [46] A. Balador, N. Ericsson, and Z. Bakhshi, "Communication middleware technologies for industrial distributed control systems: A literature review," presented at the 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA), Limassol, Cyprus, Sep. 2017.
- [47] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, "OPC UA versus ROS, DDS, and MQTT: Performance evaluation of Industry 4.0 protocols," in *Proc. IEEE Int. Conf. Ind. Technol. (ICT)*, Feb. 2019, pp. 955–962.
- [48] HiveMQ. (2020). *MQTT & MQTT 5 Essentials: HiveMQ GmbH*. [Online]. Available: <https://www.hivemq.com/mqtt-5/>
- [49] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020.
- [50] (2022). *OPC Unified Architecture Discovery and Global Services (Release 1.05.02)*. [Online]. Available: <https://reference.opcfoundation.org/GDS/v105/docs/>
- [51] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *Proc. IEEE Mil. Commun. Conf.*, vol. 1, Oct. 2003, pp. 242–247, doi: [10.1109/MILCOM.2003.1290110](https://doi.org/10.1109/MILCOM.2003.1290110).
- [52] J. Park, R. Delgado, and B. W. Choi, "Real-time characteristics of ROS 2.0 in multiagent robot systems: An empirical study," *IEEE Access*, vol. 8, pp. 154637–154651, 2020.
- [53] H. Pérez, J. J. Gutiérrez, S. Peiró, and A. Crespo, "Distributed architecture for developing mixed-criticality systems in multi-core platforms," *J. Syst. Softw.*, vol. 123, pp. 145–159, Jan. 2017.
- [54] N.-K. Hsieh, W.-Y. Lin, and H.-T. Young, "High-speed spindle fault diagnosis with the empirical mode decomposition and multiscale entropy method," *Entropy*, vol. 17, no. 4, pp. 2170–2183, Apr. 2015.
- [55] A. Abdul Kadir, X. Xu, and E. Hämmerle, "Virtual machine tools and virtual machining—A technological review," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 3, pp. 494–508, Jun. 2011.
- [56] C. Zhang et al., "A multi-level modelling and fidelity evaluation method of digital twins for creating smart production equipment in Industry 4.0," *Int. J. Prod. Res.*, Aug. 2023, doi: [10.1080/00207543.2023.2246161](https://doi.org/10.1080/00207543.2023.2246161).
- [57] D. Brandl and C. Johnsson, "Beyond the pyramid: Using ISA95 for industry 4.0 and smart manufacturing," in *InTech Magazine*. NC, USA: International Society of Automation, Oct. 2021, pp. 14–16. [Online]. Available: <https://www.isa.org/intech-home/2021/october-2021>
- [58] F. Wascher, "MES4: Communication with MES 4," Festo Didactic SE, Denkendorf, 2014.
- [59] E. Gonçalves, A. Freitas, and S. Botelho, "An AutomationML based ontology for sensor fusion in industrial plants," *Sensors*, vol. 19, no. 6, p. 1311, Mar. 2019.
- [60] C. Zhang, G. Zhou, J. Li, F. Chang, K. Ding, and D. Ma, "A multi-access edge computing enabled framework for the construction of a knowledge-sharing intelligent machine tool swarm in Industry 4.0," *J. Manuf. Syst.*, vol. 66, pp. 56–70, Feb. 2023.



**CHIN-TE LIN** was born in Kaohsiung, Taiwan, in 1979. He received the bachelor's, master's, and Ph.D. degrees in mechanical engineering from the National Taiwan University, Taipei, in 2002, 2005, and 2012, respectively.

In 2012, he joined the Industrial Technology Research Institute (ITRI). During his tenure, he completed various projects, including the Process Monitoring of CNC Machine Centers, the Industrial Upgrading of a Saw Blade Company, the Development of Software and Hardware for Machine Tool Communication Agents, and the Information Systems for Smart Manufacturing. Since 2019, he has been a Faculty Member with the Department of Mechanical Engineering, National Central University, Taoyuan, Taiwan. His current research interests include computer-aided integration in manufacturing and the application fields, includes manufacturing systems, artificial intelligence applications, production scheduling, and robotics in industry.



**HONG-JI LU** received the bachelor's and master's degrees in mechanical engineering from the National Central University, Taiwan, in 2020 and 2022, respectively.

From 2020 to 2022, he was a Teaching Assistant of various courses with the National Central University. He also assisted in research to incorporate the Internet of Things (IoT), distributed systems, and industrial communication protocols into industrial manufacturing systems, aiming to optimize production processes. Since 2022, he has been a Product Application Engineer with the Industrial IoT Business Group, Advantech Company Ltd. He is responsible for solving customer problems, proposing more feasible solutions, and promoting the practical application of smart IoT in industrial fields.

• • •