IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 9, 2017, accepted November 1, 2017, date of publication November 7, 2017, date of current version December 5, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2770180

An Integrated Industrial Ethernet Solution for the Implementation of Smart Factory

SHIYONG WANG[1], JIAN OUYANG[1], DI LI[1], AND CHENGLIANG LIU[2]
[1]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China
[2]School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Jian Ouyang (jim_oyoung@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51605168, in part by the Natural Science Foundation of Guangdong Province under Grant 2015A030308002, in part by the Science and Technology Planning Project of Guangdong Province under Grant 4092015B010917001, in part by the Fundamental Research Funds for the Central Universities under Grant 2017MS016, in part by the National Science and Technology Support Program of China under Grant 2015BAF13B00, and in part by the National Key Technologies R&D Program under Grant 2017YFD0700602.

ABSTRACT Smart factory addresses the vertical integration of physical entities and information systems. Network and cloud are two essential infrastructures to achieve this goal. Among them, the network provides interconnection for communication and data flow, while the cloud provides powerful and elastic computing and storage abilities for big data and intelligent applications. This paper presents a cloud-centric framework for the implementation of the smart factory. Based on this framework, three high leveled protocols viz., EtherCAT, DDS, and OPC UA are selected to implement machine network (controller to sensors/actuators communication), machine to machine communication, and machine to cloud communication respectively, to satisfy diverse communication requirements of the smart factory. An integrated architecture for combining ontology knowledge and semantic data to support intelligent applications is also proposed. These network and data process schemes are verified in a smart factory prototype for personalized candy packing application.

INDEX TERMS Smart factory, Industry 4.0, OPC UA, DDS, EtherCAT.

I. INTRODUCTION

The development of industry is marked by various big events. The invention and wide application of stream engines resulted in the world's first industrial revolution at the end of the 18^{th} century. The second industrial revolution began around the turn of the 20^{th} century when electric motors were created to replace stream engines in most domains. We are currently in the third industrial revolution which started in the early 1970s. Automated machines enabled by programmable NCs and PLCs, and information systems such as MES and ERP are deployed to build the automation pyramid. Now, the fourth industrial revolution, industry 4.0 [1], is expected with the significant advancements in information technologies and artificial intelligence technologies, including but not limited to cloud computing [2], big data [3], mobile internet [4], and machine learning [5]. Along with the technological advancements, come the practical needs. Personalized consumption, environmental pollution, and population aging ask for flexible, green, and human-friendly production paradigm.

Industry 4.0 identifies three key features, viz., (1) the vertical integration, (2) the horizontal integration, and (3) the end to end engineering integration. The smart factory addresses the vertical integration. From the technology point of view, cloud and network are the two essential infrastructures to support the smart factory, because almost all the components such as materials, machines, products, and people should be connected and the generated mass data should be transferred, stored, and processed. High bandwidth networks are for connection and data transfer, and the cloud is for data storage and processing. As users, one can imagine a very simple model of network and cloud. The network acts as a channel, and every network node is equipped with a required interface to transfer the data. The cloud is a supercomputer to provide computing and storage resources on demand. However, realizing such a simple model needs much more efforts [6].

A test-bed called SmartFactory^{KL} was established in Germany, and several wireless communication systems were deployed in the demonstration facility [7]. The wireless communication is flexible compared to the wired solutions, but it still lacks robust industrial wireless communication products. Wang [8] *et al.* used the general Ethernet/Internet connection for the machine to machine communication and machine to cloud communication. Although the Internet is widely used, it should be tuned to accommodate industry specific requirements such as real-time processing and security. Fortunately, this problem gained much more attention over time, and multiple industrial Ethernet standards have been proposed. Among these, EtherCAT aims to provide high real-time processing capabilities [9] and synchronization [10], with a higher bandwidth utilization. Data Distribution Service (DDS) [11] is for peer communication with guaranteed QoS. The OPC Unified Architecture (OPC UA) [12] features information modeling and secure communication. These standards were proposed for different applications, and a combination of them may provide a practical solution for the complicated factory application.

In this paper, we aim to provide a network solution for the smart factory. Our motivation is to set up a flat network topology while satisfying the heterogeneous requirements from different participants and layers, in terms of bandwidth, reliability, real-time processing, security, and semantics. In the proposed scheme, Ethernet is used as a base network technology, and different high-level protocols are overlapped to meet the specific requirements. Ethernet is logically a bus network equipped with switchers, which can be physically deployed as a star or tree or hybrid network. Therefore, Ethernet enables a very convenient and easy connection, and network nodes are peers. Such a connection is quite time-saving and economical while implementing the smart factory. Standard industrial Ethernet protocols (high-level protocols) including EtherCAT, DDS, and OPC UA are explored and implemented above the Ethernet. Note that the general purpose Internet protocols can co-exist with these industrial protocols. Based on this network solution, mass semantic data can aggregate to the cloud, so we propose a scheme for intelligent industrial big data analytics running on the cloud.

The paper is organized as follows. The framework of the smart factory is set up, and various layers are explained in Section II. We presented the highlights and application schemes of EtherCAT, DDS, and OPC UA to the three layers in Sections III, IV, and V, respectively. As a natural extension, we proposed a processing architecture to integrate knowledge with data for intelligent applications in Section VI. In Section VII, we introduced a prototype smart factory and described how the abovementioned technologies are used. Finally, conclusions are drawn in Section VIII.

II. FRAMEWORK AND REQUIREMENTS

This section presents a cloud-centric framework for the smart factory followed by a summary of three kinds of communication technologies existing at different levels.

A. SMART FACTORY FRAMEWORK

When thinking of the cloud as the brain of the smart factory, the network can be considered as the blood vessels and nerve cells to connect the shop-floor entities and client terminals as the body to the brain. In this way, we capture a cloud-centric framework of the smart factory as shown in Fig. 1. From the hardware point of view, the smart factory features a flat communication architecture where the shop-floor entities, the cloud, and the client terminals are connected together.

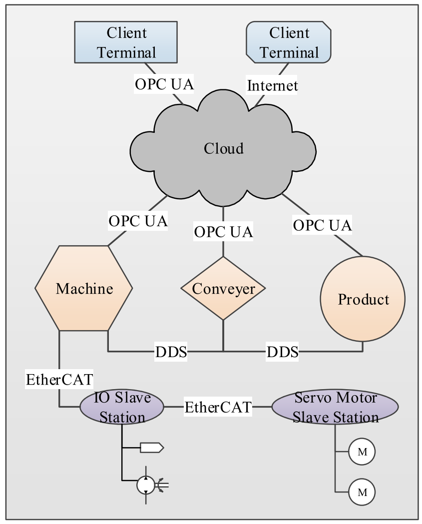


FIGURE 1. Cloud-centric framework of smart factory.

The cloud is functionally equivalent to a supercomputer with extendable computing and storage abilities, enabling it a suitable solution to host ever increasing industrial data and diverse application software. The shop-floor entities such as machines, conveyors, and products are autonomous agents that can interact with each other and with the cloud. The client terminals such as computers and smartphones can also be viewed as intelligent agents to assist people to observe and operate the smart factory.

The above smart factory framework differs from the conventional factory architecture in three aspects. First, the cloud is used instead of separate servers. Second, a single network technology, e.g., Ethernet and Internet, replaces the diverse protocol and spreads from control network to operation network till corporate network. Third, the shop-floor entities and client terminals are generalized as intelligent agents. These agents can make decisions, interact with partners, perceive their surrounding environment, and perform physical actions. Therefore, the smart factory is characterized by a uniform cloud, network, and agent model that can be used in various concrete applications.

B. THREE LAYERED NETWORKS

Fig. 1 shows the communication that can occur at different levels: within a machine (between the controller, sensors, and actuators), inter machines, and between machines and the cloud. At present, Ethernet can be used in all these levels enabling a uniform network. However, to provide different QoS (TABLE I), different high protocols should be used at these levels. For example, EtherCAT can be used in the controller to sensors/actuators network communication to achieve high real-time processing and synchronization; DDS can be used in the machine to machine network communication for high bandwidth and proper performance;

TABLE 1. Relative performance requirements of different network levels.

	Bandwidth	Reliability	Real-time	Security	Semantics
Controller to sensors and actuators network	★	★★★	★★★	★	★
Machine to machine network	★★	★★	★★	★★	★★
Machine to cloud network	★★★	★	★	★★★	★★★

OPC UA is suitable for the machine to cloud network communication to provide semantic information and high security.

III. MACHINE NETWORK FOR THE CONTROLLER CONNECTING ITS SENSORS AND ACTUATORS

A. EtherCAT

Ethernet using the competitive CSMA/CD media access mechanism makes the time determinacy impossible. There are solutions trying to disable the CSMA/DA via higher level protocol layers and using the time slicing procedure or polling method instead. However, for process control applications, the usable data rate is quite low because of a constant Ethernet header and tailer (84 bytes per frame). This is because the master communicates with each of its slaves using individual frames. EtherCAT overcomes this problem by building a logical ring of slaves and sending a frame flowing through the ring. During the flow, each slave can retrieve its command and insert its feedback to the frame.

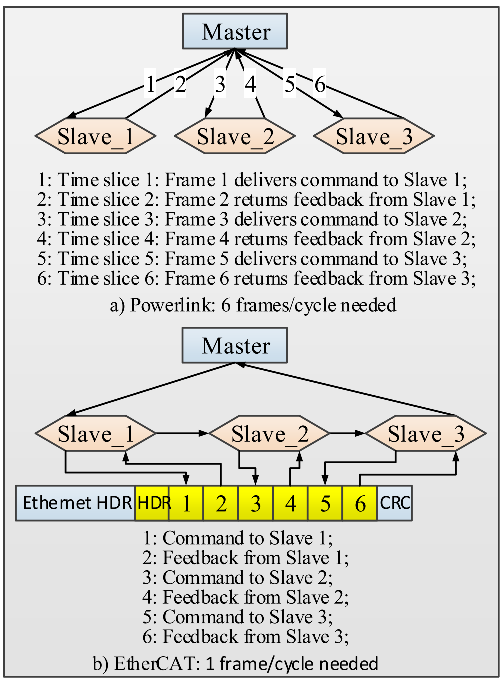


FIGURE 2. Comparison of industrial Ethernet using time slice and EtherCAT.

As illustrated in Fig. 2, the Powerlink (based on time slice) uses six frames to communicate with its three slaves, to send commands and receive feedbacks. Assuming that the command and feedback need four bytes each, the maximum usable data rate is only $4/(84+4) = 4.55\%$, regardless of the number of slaves. For the EtherCAT solution, only one frame is sufficient to realize the same function, and the usable data rate is about $(8^*3)/(84 + 8^*3) = 22.22\%$, where there are eight bytes for each slave. With an increase of the quantity of slaves, the usable data rate can be over 90%.

B. APPLYING EtherCAT IN SMART FACTORY

The EtherCAT has both the advantages of determinacy and high speed which results in real time. Moreover, an accurate synchronization is also available due to the distributed clock technology. The characteristics of real-time and synchronization make the EtherCAT not only able to replace the conventional field buses to connect multiple salves but the distributed control of machines needing IO processing and synchronous servo control as well.

IV. INTER-MACHINE NETWORK FOR PEER-TO-PEER NEGOTIATION

This section describes the scheme to apply DDS for machine-to-machine communication, including abstract communication process and practical network configuration.

A. DDS

DDS specifies a standard publish-subscribe communication model for a heterogeneous group of entities where the participants are not classified into servers and clients. Fig. 3 describes the main components and characteristics of a system communicating via DDS. An application using DDS for communication is called DomainParticipant. Multiple domains (distinguished from each other by domain IDs) can exist on the same network, but a DomainParticipant can only join one of them. A DomainParticipant can be either a Publisher or a Subscriber or both. A Publisher contains one or more DataWriters, while a Subscriber manages one or more DataReaders.

A DataWriter or a DataReader is bound to a Topic at the creation time from its Publisher and Subscriber, respectively. The Topic has a name and a type (describing a complex or simple data structure), and the name is unique in the domain. Note that a DataWriter and a DataReader are interconnected logically if they are bound to the same Topic. However, a Subscriber need not know where the Publisher is located. Multiple Publishers can publish to the same Topic, and also multiple Subscribers can subscribe the same Topic. Two DomainParticipants that are implemented as both the Publisher and Subscriber can communication with each other in a bidirectional way.

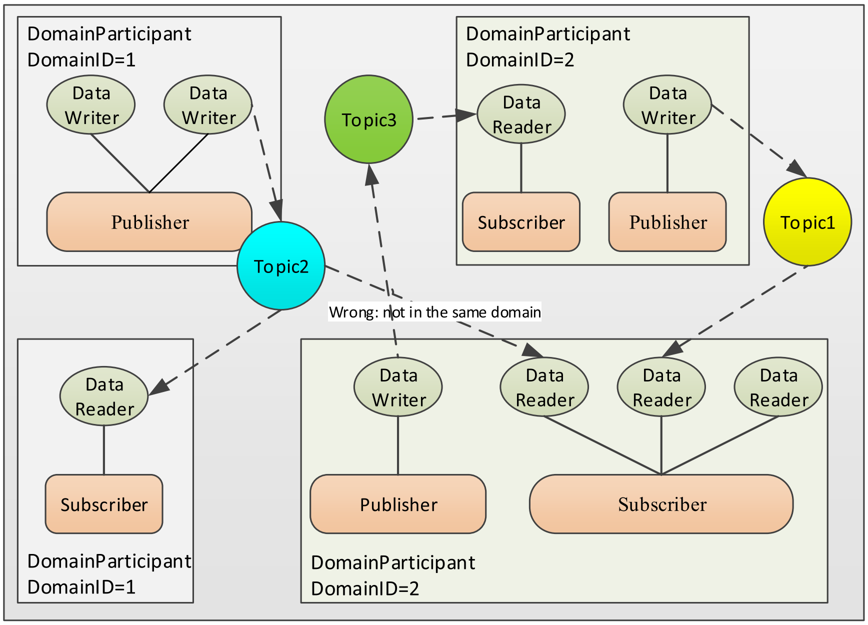


FIGURE 3. Main components and characteristic of DDS.

B. APPLYING DDS IN SMART FACTORY

In the smart factory, products, machines, and conveyors can negotiate with each other to organize resources in a self-organized way, i.e., without a central controller. Wang *et al.* [13] proposed an intelligent negotiation mechanism based on the Contract Net Protocol (CNP). The product will act as the manager to negotiate with machines for operations and with conveyors for constructing routes to transport the product among machines. CNP defines six message type for communication between a manager and contractors. The manager uses the ANNOUNCE message to publish the required operations; contractors use the BID message to bid for the operation; finally, the manager uses the AWARD or DIRECTEDAWARD message to confirm the winning contractor. Also, the manager can use the REQUEST message to ask for information, and the contractors can respond with the INFORM message. The main configurations for applying the DDS in the smart factory to facilitate the CNP based negotiation are shown in Fig. 4.

Two domains are defined, one for determining the machines to perform operations called MachiningDomain, the other for determining the conveyor to transport products called ConveyingDomain. Machines and conveyors are in the MachiningDomin and the ConveyingDomain respectively, but products should be in both domains because the products are managers that organize communication in both domains. Each machine or conveyor has an application implemented as both Publisher and Subscriber, while each product has two applications, one in the MachiningDomin and the other in the ConveyingDomain, both are Publisher and Subscriber.

In the MachiningDomain, three topics are defined for each operation corresponding to the ANNOUNCE, BID, and AWARD. For example, topics OP1_ANNOUNCE, OP1_BID, and OP1_AWARD will be defined for the operation OP1. On one hand, operations may have different performance requirements which demand different data structures to express. On the other hand, regardless of the number of operations and machines, a single operation is generally supported by a very limited number of machines, so defining Topics based on the operations can effectively restrict the number of participants bound to a Topic. In the ConveyingDomain, only a REQUEST topic and an INFORM topic are defined, as the negotiation process does not relate to the operations.

The number of DataReaders and DataWriters of a product is equal to the sum of the number of operation types required by the product, one DataWriter for the REQUEST topic and one DataReader for the INFORM topic. The number of DataReaders and DataWriters of a machine is equal to the number of operation types supported by the machine. Each conveyor has only one DataWriter and one DataReader corresponding to the REQUEST topic and an INFORM topic.

V. SEMANTIC NETWORK FOR DATA ACQUISITION AND CLOUD ASSISTANCE

This section presents the application of the OPC UA specification in a smart factory for standard information expression and exchange. These data related schemes are independent of concrete production scenarios and suitable for a range of smart factory applications.

A. OPC UA

OPC UA specification is a revolutionary evolution of OPC standard to provide interoperable, platform-independent,

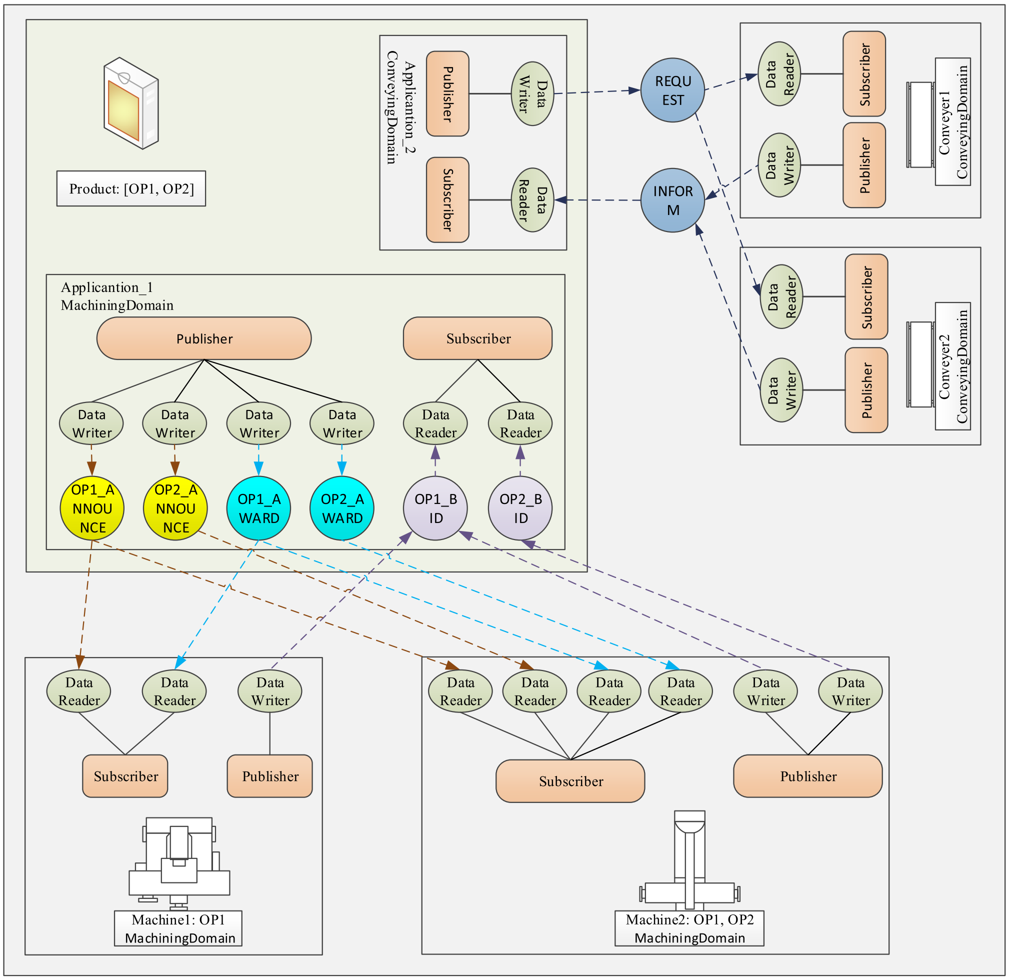


FIGURE 4. Practical configuration of DDS for negotiation via CNP.

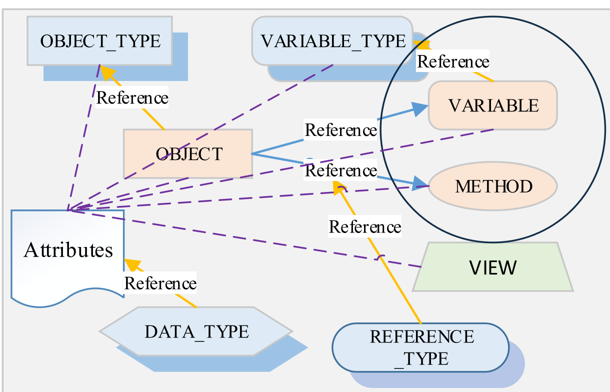


FIGURE 5. Functions of and relationship between nodes of different node classes.

high-performing, scalable, secure, and reliable communication between applications. It mainly refers to information modeling and data transport. As shown in Fig. 5, OPC UA uses nodes to represent information. There are a total of eight node classes defined to express different data structures.

Node classes OBJECT, VARIABLE, and METHOD are used to code the data in an object-oriented method, i.e., an object uses variables to express properties and methods to express supported actions. Node classes OBJECT_TYPE, VARIABLE_TYPE are used to express the type information, in other words, the semantics of OBJECT and VARIABLE respectively. The node class REFERENCE_TYPE defines the semantics of references that connect one node to another, and DATA_TYPE describes the semantics for attributes. The node class VIEW can be used to organize the data space by providing an entrance to a subset of information that one is interested in. An object connects to its variables and methods using HasComponent references, and a node connects to its type using HasTypeDefinition reference. A reference is not a node, but it is a complex data structure storing the name of the reference type in one of its members. All node classes specify a fixed number of attributes, each of which has a data type.

B. APPLYING OPC UA IN SMART FACTORY

In the smart factory, the shop-floor entities can be implemented as OPC-UA servers to provide semantic data to the

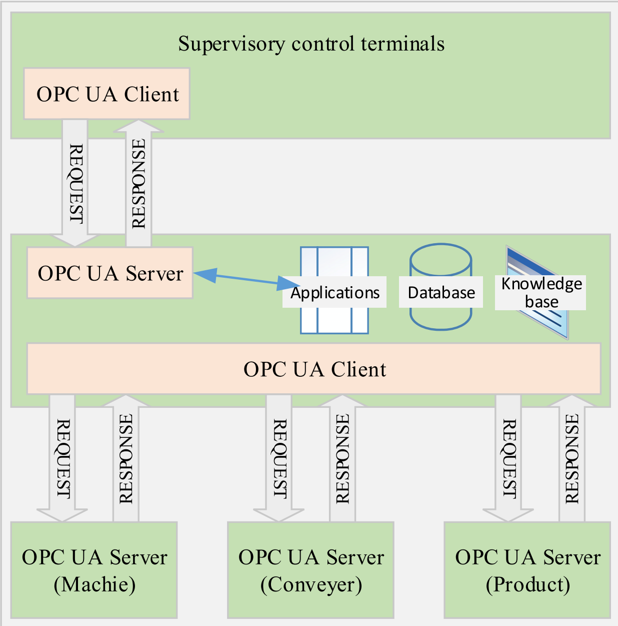


FIGURE 6. OPC-UA system architecture for the smart factory.

cloud that is acting as an OPC-UA client (Fig. 6). This client on the cloud can deliver real-time data and events to applications, and historical data and events to the database. An application can be optionally equipped with an OPC-UA server to interact with the supervisory control terminals acting as OPC-UA clients.

VI. INTEGRATING KNOWLEDGE WITH DATA FOR INTELLIGENT APPLICATIONS

Fig. 7 shows a solution to integrate the knowledge with data based on the underlying support from OPC-UA servers (embedded in shop-floor entities). Ontology models the objects and the relationship between objects. Establishing ontology for a smart factory application contributes not only for OPC-UA information modeling but also knowledge-based reasoning and automation. Modeling tools for building ontology like Protégé use similar concepts, such as objects and references (describing the relationship between objects) with OPC-UA. Therefore, OPC-UA information model can be easily built based on the ontology. With the rules defined in Protégé or/and Jena and real-time values from database assigned to properties of ontology objects, applications can use reasoning engine like Jena to verify the requirements and figure out violations.

In summary, OPC-UA servers upload the semantic data that may be stored in a database. The applications can directly retrieve data from the database and apply data analysis or data mining. Alternatively, the applications can also rely on reasoning engine and ontology to make intelligent decisions. Of course, both methods can be used together in a single application. In this way, knowledge and data are tightly integrated to lay a solid foundation for the smart factory applications.

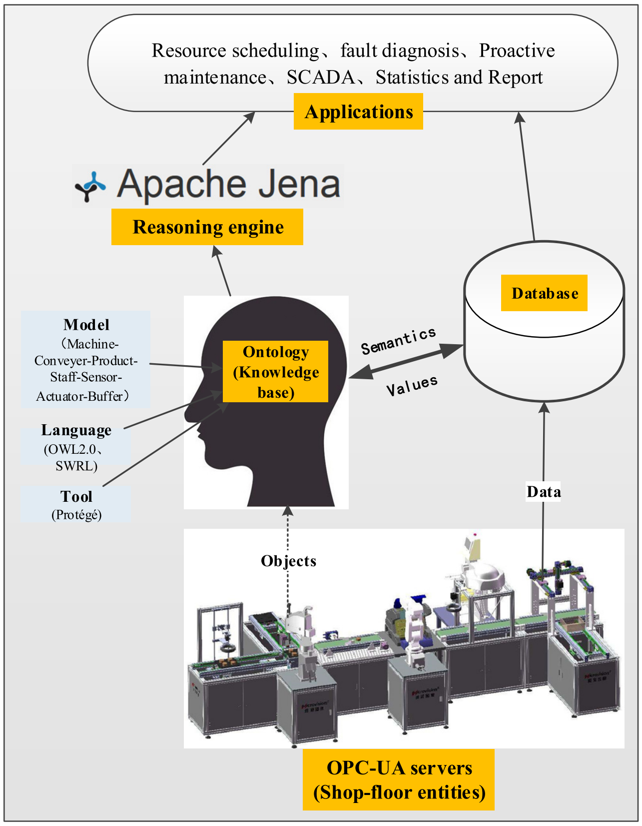


FIGURE 7. Integration of knowledge with data for smart factory.

Note that different applications utilize a uniform database and are based on a uniform ontology. Also, applications, as well as the data and knowledge base exist in the same cloud, making the deployment and interaction very convenient.

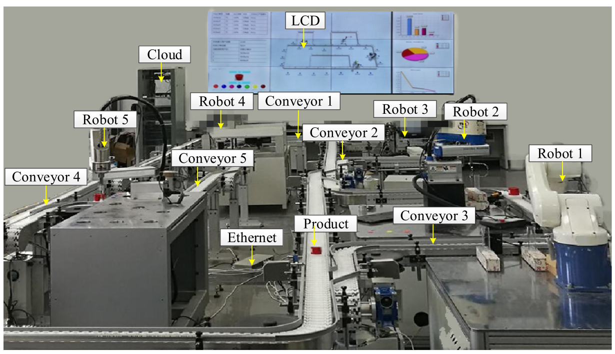


FIGURE 8. Snapshot of the prototype smart factory [14].

VII. PROTOTYPE PLATFORM

A smart factory prototype is built to simulate the personalized candy packing as shown in Fig. 8 [14]. A consumer can customize the box types, the candy types and the quantity of each selected candy type. After the order is sent to the smart factory, the needed robots and routes are dynamically determined based on the intelligent negotiation mechanism. Candy boxes and conveyor belts are products and conveyors respectively. The robots for loading candies, moving boxes to the conveyor belts, and downloading finished packets from the conveyor belts are machines.

A. SMART MACHINES DESIGNED WITH OPC UA, DDS, AND EtherCAT

We have developed a smart controller that supports OPC UA, DDS, and EtherCAT, and applied it to the SCARA Robot 1. It involves a master and slaves. The master is an industrial personal computer (IPC) that is used to host the OPC UA server, the DDS node, and the EtherCAT master. The IPC uses separate Ethernet interfaces for each protocol to improve performance. The EtherCAT slave is an embedded controller powered with Beckhoff ET1100 chip, supporting eight servo axes, 32 digital input ports, and 32 digital output ports. As one slave controller can control multiple axes, complex trajectories can be interpolated. This design is suitable for machine control, as the servo axes and IOs of a machine are close to each other. Additional slaves can join in the network when more servo axes and IOs are needed.

B. LEGACY MACHINES SUPPLEMENT WITH CONTROLLERS TO SUPPORT OPC UA AND DDS

Legacy machines such as industrial robots purchased from the third party vendors and PLCs are the closed systems; new protocols are not allowed to be added even if the legacy controller has an Ethernet interface. In the prototype system, a 6-axis robotic arm (VS-6577G, DENSO Corp, Japan), a SCARA robotic arm (THL300, TOSHIBA Corp, Japan), two Mitsubishi PLCs for controlling box uploader and box downloader, and five Siemens PLCs for controlling conveyor belters are used. The PLCs connect its IOs and servo axes using proprietary signal lines. Therefore, EtherCAT is not needed. To support OPC UA and DDS, a Raspberry Pi microcontroller is used and deployed along a legacy machine or PLC to take over the responsibility of negotiation and communication. The Raspberry Pi microcontroller communicates with its client via ordinary Ethernet.

VIII. CONCLUSION

Network and communication are fundamental requirements for the implementation of the smart factory. Ethernet draws much attention from the industrial community due to its high bandwidth, and new protocols are developed and standardized to address the specific requirements of the industrial applications. In the smart factory application, EtherCAT suits high-speed real-time synchronous control to replace the conventional field buses, e.g., in a machine network to implement the controller to sensors/actuators communication. DDS specifies a standard publish-subscribe communication model, characterized by high flexibility and suitable for the machine to machine communication. OPC UA defines a standard and extensible information modeling method, enabling the semantic data interaction between machines and cloud. These protocols rely on the same Ethernet significantly reducing the effort in terms of network construction especially in large factories. And, this solution builds a base for the integration of knowledge with semantic data that support intelligent process of industrial big data.

REFERENCES

[1] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of Industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, 2016, Art. no. 3159805.

[2] X. Xu, "From cloud computing to cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, 2012.

[3] J. Wan *et al.*, "A manufacturing big data solution for active preventive maintenance," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.

[4] D. Zhang, J. Wan, Q. Liu, X. Guan, and X. Liang, "A taxonomy of agent technologies for ubiquitous computing environments," *KSII Trans. Internet Inf. Syst.*, vol. 6, no. 2, pp. 480–497, 2012.

[5] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data management challenges in production machine learning," in *Proc. ACM Int. Conf. Manage. Data*, May 2017, pp. 1723–1726.

[6] D. Li, J. Wan, A. V. Vasilakos, C. F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless Netw.*, vol. 23, no. 1, pp. 23–41, 2017.

[7] D. Zuehlke, "SmartFactory-towards a factory-of-things," *Annu. Rev. Control*, vol. 34, no. 1, pp. 129–138, 2010.

[8] S. Wang, J. Wan, M. Imran, D. Li, and C. Zhang, "Cloud-based smart manufacturing for personalized candy packing application," *J. Supercomput.*, pp. 1–19, 2016. [Online]. Available: https://doi.org/10.1007/s11227-016-1879-4

[9] D. Jansen and H. Buttner, "Real-time Ethernet: the EtherCAT solution," *Comput. Control Eng.*, vol. 15, no. 1, pp. 16–21, 2004.

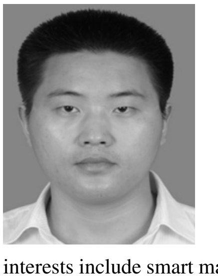
[10] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "Evaluation of EtherCAT distributed clock performance," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 20–29, Feb. 2012.

[11] C. Esposito, S. Russo, and D. D. Crescenzo, "Performance assessment of OMG compliant data distribution middleware," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–8.

[12] G. Cândido, F. Jammes, J. B. de Oliveira, and A. W. Colombo, "SOA at device level in the industrial domain: Assessment of OPC UA and DPWS specifications," in *Proc. 8th IEEE Int. Conf. Ind. Informat.*, Jul. 2010, pp. 598–603.

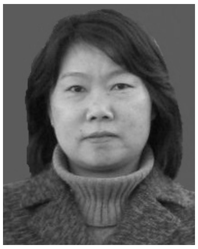
[13] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination," *Comput. Netw.*, vol. 101, pp. 158–168, Jun. 2016.

[14] D. Li, H. Tang, S. Wang, and C. Liu, "A big data enabled load-balancing control for smart manufacturing of Industry 4.0," *Cluster Comput.*, vol. 20, no. 2, pp. 1855–1864, Jun. 2017.



SHIYONG WANG was born in Huoqiu County, Anhui, China, in 1981. He received the B.S. and Ph.D. degrees in mechanical and electrical engineering from the South China University of Technology, Guangzhou City, China, in 2010. From 2010 to 2016, he was a Lecturer, and is currently an Assistant Professor with the Mechanical and Electrical Engineering Department, South China University of Technology. He has authored over 20 articles and holds five patents. His research interests include smart manufacturing, motion control, and robotics. He was a recipient of the First Prize for Science and Technology Development of Guangdong Province in 2009.

JIAN OUYANG received the B.S. degree from Shijiazhuang Tiedao University, Heibei, China, the M.E. degree from Southwest Jiaotong University, Sichuan China, and the Ph.D. degree in mechanical and electrical engineering from the South China University of Technology, Guangdong, China. Since 2016, he has been an Assistant Researcher with the College of Computer Science and Engineering, South China University of Technology. He has authored over ten articles and holds seven patents. His current interests include battery management system, intelligent control, and electric vehicles.



DI LI is currently a Professor with the School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China. She has directed over 50 research projects, including the National Natural Science Foundation of China. She has authored/co-authored over 180 scientific papers. Her research interests include embedded systems, computer vision, and cyber-physical systems.

CHENGLIANG LIU received the B.E. degree from the Shandong University of Technology, Shandong, China, and the M.E. and Ph.D. degrees from Southeast University, Nanjing, China, in 1991 and 1998, respectively. He has been invited as a Senior Scholar at the University of Michigan, Ann Arbor, and the University of Wisconsin-Madison, Madison, since 2001. He is currently a Professor with the Department of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China. His current interests include mechatronic systems, intelligent maintenance, MEMS design, intelligent robot control, remote monitoring techniques, and condition-based monitoring.