IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal
SPECIAL SECTION ON KEY TECHNOLOGIES FOR SMART FACTORY OF INDUSTRY 4.0

Received September 11, 2017, accepted September 27, 2017, date of publication September 29, 2017, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2758160

CASOA: An Architecture for Agent-Based Manufacturing System in the Context of Industry 4.0

HAO TANG[1], DI LI[1], SHIYONG WANG[1], AND ZHIJIE DONG[2]
[1]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China
[2]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Zhijie Dong (dongzj@scut.edu.cn)

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 2015B010917001, and in part by the Fundamental Research Funds for the Central Universities under Grant 2017MS016.

ABSTRACT The fourth industrial revolution involves the advanced topics, such as industrial Internet of Things, cyber-physical system and smart manufacturing that address increasing demands for mass customized manufacturing. The agent-based manufacturing is a highly distributed control paradigm that can cope with these challenges well. This paper gives an overview of agent-based architectures for manufacturing systems. Besides, a cloud-assisted self-organized architecture is presented by comprising smart agents and cloud to communicate and negotiate through networks. Ontological representations of knowledge base are constructed to provide the information basis for decision-making of agents, which enables dynamic reconfiguration among agents in a collaborative way to achieve agility and flexibility. Furthermore, the agents' interaction behavior is modeled to structure the agents hierarchically to reduce the complexity, because the interactions among agents in distributed system are difficult to understand and predict. The experimental results show that the presented architecture can be easily deployed to build smart manufacturing system and can improve the adaptiveness and robustness of the manufacturing system when dealing with mixed multi-product tasks.

INDEX TERMS Multi-agent system, industry 4.0, ontology, smart manufacturing.

I. INTRODUCTION

The current market demands are becoming much tougher and more severe, especially on product customization, small batch and low prices [1]. The challenges require a new balance among economy, technology and society [2]. Traditionally, product lines are designed in a rigid pattern to produce limited types of products. What's worse, reconfiguring a legacy production line or building a new one from scratch is costly and time-consuming [3]. To survive in this new manufacturing environment, enterprises in this industry must be able to react to rapid changes with reconfigurability [4]. This necessitates the needs of new manufacturing methods which are capable of managing continuous product changes and disturbances efficiently and reliably [5], [6].

A number of technologies and design approaches have been presented to improve reconfigurability in manufacturing systems, including modular machine tools and reconfigurable automation during the last two decades [7], [8]. Several IT-driven paradigms such as Multi-Agent Systems (MAS), Holonic Manufacturing Systems (HMS), Evolvable Assembly Systems (EAS), and Fractal Factories (FF) have emerged. In addition, many notable reference architectures like reference architecture for holonic manufacturing systems (PROSA) and a holonic architecture for agile and adaptive manufacturing control (ADACOR) were proposed [9]–[16]. Although the technological advances are able to achieve reconfigurability theoretically, their ultimate industrial implementation remains restrained. The implementing difficulty is short of quantitative multi-agent system design approaches based on reconfigurability measurement.

An agent-based manufacturing architecture can ease self-configuration, modification of the system and enable a larger decision space for reacting to external environment [17].

Thus, it is important to build a knowledge base to provide the information basis for decision-making [18]. An ontological information modeling method is put forward to build the knowledge base, which integrates, analyzes and processes enormous manufacturing information and extracts answers on the basis of semantics [19]–[21]. The MAS defines autonomous and smart entities as agents, which are able to collaborate with each other to complete global tasks [22]. The basic need for cooperation rises from the fact that agents don't have enough knowledge to make global decisions. As a result, we introduce the cloud-assistant mechanism to coordinate the agents globally. However, the interaction behavior of agents in a distributed system is difficult to understand and predict. Therefore, the study of behaviors among multiple agents is used to structure the agents in a hierarchy to reduce the complexity.

This paper describes a distributed architecture for agent-based manufacturing system, and the contributions are as follows. Firstly, the ontological knowledge base is established to enable agents' cooperation in order to complete the tasks. Secondly, the interaction behaviors among agents are proposed to explain the dynamic scheduling schemes in three scenarios. Thirdly, two communication methods are introduced to meet the communicated demands for interoperability in smart manufacturing. Lastly, the superiorities of proposed architecture are verified and validated by a proof-of-concept experiment.

The article is organized as follows. In Section 2, an overview of existing manufacturing architectures is given and knowledge-driven interaction behaviors among agents are discussed. In Section 3, two communication methods used in the proposed architecture are described. In Section 4, the experimental setup and analysis of main experimental results are provided. Finally, Section 5 concludes the paper.

II. AGENT-BASED CONTROL STRUCTURE

The core principle of smart manufacturing is the adoption of Internet of Things (IoT), or in other words connecting of people, data, and things through networks. Besides, in a smart factory, the prime consideration is to build a vertically integrated system for data consolidation from Enterprise Resource Planning (ERP) systems to plants. This requires a multi-layer network that makes data identifiable across different layers. Therefore, a smart manufacturing architecture called cloud-assisted self-organized architecture is presented to realize the consideration. As shown in Fig. 1, CASOA comprises two layers: the lower resource layer and the upper cloud-assistant layer.

The lower resource layer that contains devices is the basis of IIoT. This layer enables smart objects to collaborate with each other to complete production tasks without a centralized scheduler. According to various manufacturing tasks, this way of distributed work makes plants dynamically coordinated in order to meet product requirements with small batch and variety. To eliminate potential local optima in the case of fully distributed scheduling, the upper cloud-assistant layer is introduced. The upper cloud-assistant layer collects data from the lower resource layer and figures out optimal scheduling plan through data analysis. The plans as suggestions are fed back to the plants for assisted scheduling. In the upper cloud-assistant layer, the Supervisory Control and Data Acquisition system (SCADA) and Manufacturing Execution System (MES) blend together: the SCADA system extends upward for more rich abilities of service and data management; the MES extends downward for more capacities of communication with plants. These two systems cooperate with ERP system to acquire the functions such as resource scheduling, task management, data processing and visualization. Thus, compared to the traditional manufacturing architecture, CASOA is both standard and more flexible with highly customizable features.

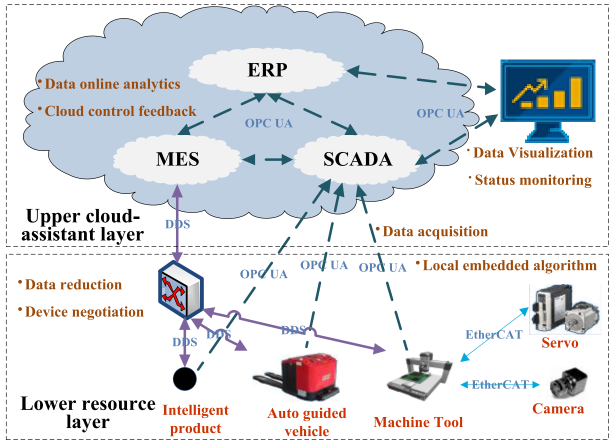


FIGURE 1. The cloud-assisted self-organized architecture.

The lower resources of CASOA compete and cooperate in a decentralized manufacturing context with self-organized ability. They contribute to complementary advantages to achieve flexibility when processing multi-type products. In addition, the components of the cloud can infer suggestion plans for optimization of lower resources scheduling. This is an organic convergence of distributed and centralized systems that exists the natural relation with MAS. Therefore, the smart entities in CASOA can be modeled as MAS to build a smart manufacturing system based on agents. A solution for the smart manufacturing can be presented by studying the interaction behaviors among agents.

A. BASIC AGENTS

The smart entities in CASOA can be classified into four types of agents according to their functions and responsibilities: suggestion agents (SA), product agents (PA), machining agents (MA) and conveying agents (CA).

The product agents represent the products that need to be processed. In contrast to the traditional products, the product agents as autonomous and collaborative nodes are able to interact with other agents. So, the product agents should be equipped with some communication tools which can

TABLE 1. Three types of communication tool.

Communication tool	Price	Speed	Read/Write
Controller	★★★	★★★	Both
RFID	★★	★★	Both
QR/bar code	★	★	Read

be divided into three types: embedded controllers, radio frequency identification devices (RFIDs) [23], and Quick Response (QR)/bar codes. As shown in TABLE 1, the data transmission speed of embedded controller is the fastest and embedded controller can both read and write data. But it is the most expensive communication tool. The data transmission speed of RFID is moderate and data can be both read and written. Meanwhile, the price is also moderate. The data transmission speed of QR/bar code is the slowest and data can be only read, and the price is the lowest.

The machining agents can perform machining and storing of operations, referring to machine tools, testing equipments and so on. The conveying agents can transport the product agent from current location to the destination, referring to conveyor belts, auto guided vehicles and so on. The latter two kinds of agents are the basic manufacturing units. Each machining agent can complete one or more processes. Due to the task complexity, every task needs the engagement of various machining agents. Similarly, the transportation of product agents also requires several conveying agents to negotiate and cooperate for finishing the task.

The suggestion agent represents the software component on the cloud which is responsible for processing orders and generating scheduling suggestions. The traditional central scheduler produces scheduling commands to directly allocate tasks to plants. However, in the lower resource layer of CASOA, the plants emphasize dynamic autonomous scheduling and are able to negotiate to allocate the tasks. Hence, the suggestion agent assists lower resources in dynamic scheduling rather than directly allocates the tasks. For instance, when a load-unbalancing of plants occurs, the suggestion agent considers the unbalancing degree to adjust scheduling plans in the suggested way [2].

B. ONTOLOGY COMBINED WITH AGENTS

In MAS, an agent is able to produce plans and make autonomous decisions to react to the external environment. Therefore, the first thing to consider is how the agents generate application plans. Ontology techniques are introduced in order to provide the theoretical basis to build the knowledge base model of the agent. Ontology is a philosophy conception which systematically describes objective things on the earth. It aims to capture the knowledge of relevant field and define the objects, which forms the semantic basis for interactions of different agents. The ontology consists of class, property, constraint and instantiation of class. The object in CASOA is abstractly described and combines its property to form an ontology model. The constraints of the ontology model

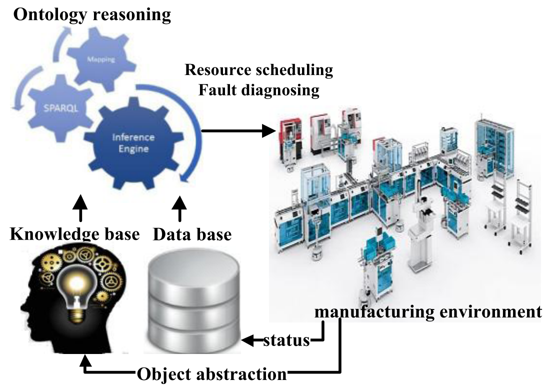


FIGURE 2. The ontology reasoning based on knowledge base and data base.

are added into different manufacturing scenarios to build the semantic knowledge base (hereafter referred to as the knowledge base).

The construction of unified knowledge base is significant to achieve interoperability among agents. The superposition of multi-agent ontology models constitutes the ontology model of the whole smart factory, which can be used to reason out the health status of the product line. As shown in Fig. 2, the status data are uploaded from plants to a database. Meanwhile, plants in the manufacturing environment are abstracted to build the ontology model. Then, the status data of the database are mapped to the ontology property of the knowledge base. In this way, the ontology model is associated with the status of plants. Finally, the reasoning engine produces strategies which will be applied in resource scheduling and fault diagnosis.

According to the modeling principles, a machining agent model can be obtained as presented in Fig. 3. The model consists of mechanical modules, control modules, electrical modules and tool modules. Also, these modules are divided into circuits, various tools and so on. This means the objects in the lower level are parts of the objects in the upper level, so the object property "*Hascomponent*" (Hc) can be used to indicate the relationship between lower and upper level. In addition, objects have their own data property which represents the quantifiable property of objects. For example, a motor has the data property RPM denoting its current speed. Before the machining agent accepts a task, its status should be checked to judge whether it has the processing capacity or it is broken down. Thus, the following constraint rules should be obeyed:

1) If MillTool && abrasion loss > "15%", Then MachineCondition = "*MillingFault*"
2) If MchineCondition = "*MillingFault*", Then TaskCondition = "*No accept Milling task*"

The rule 1 shows that, if the abrasion loss of mill tool is above 15%, the machining agent is in the condition of milling fault. The rule 2 shows that, if the machine condition is "*milling fault*", the machining agent will not accept the

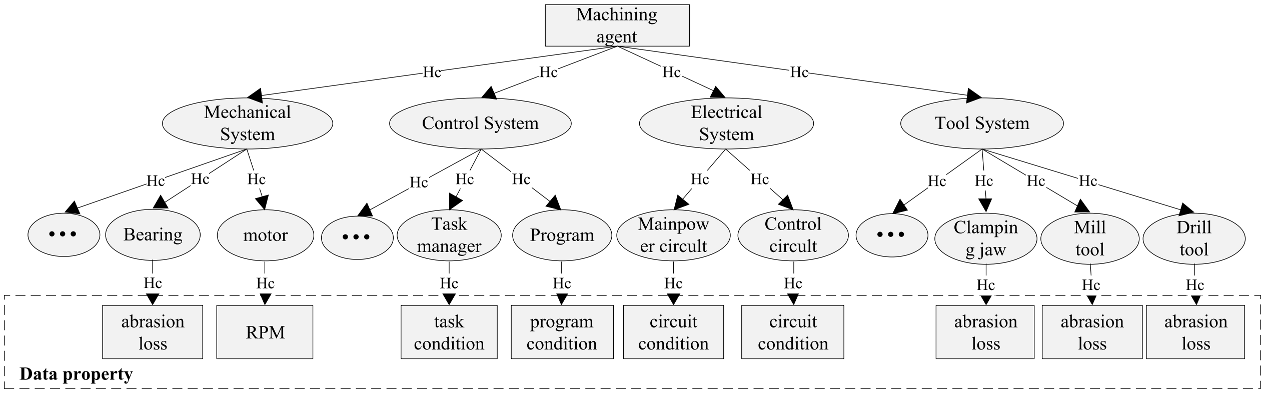


FIGURE 3. The ontology model of MA.

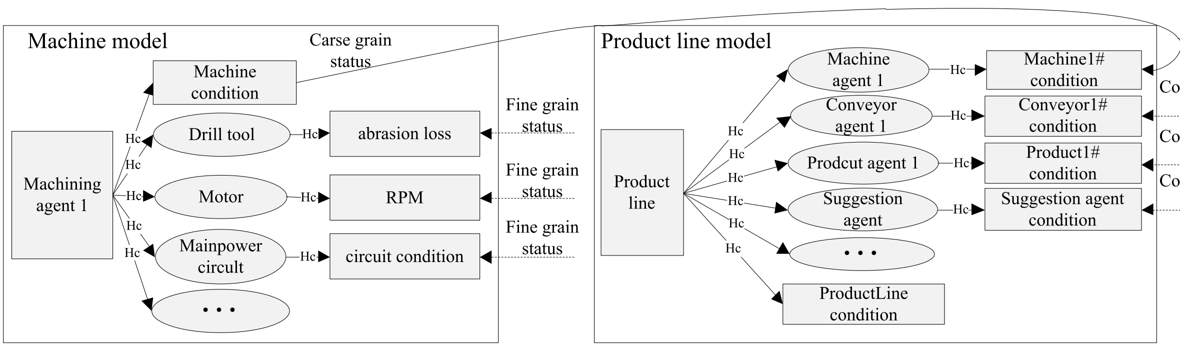


FIGURE 4. The hierarchical model of CASOA.

milling task. Therefore, the rule 1 is used in fault diagnosis and the rule 2 is used in negotiation scheduling among agents.

There are two levels of the ontology model in CASOA. The first level is the agent model whose input is the minimum level of status called fine grain status, and the output is a coarse grain status that represents the agent condition. The second level is the product line model. Whether the product line can make a product is determined by the status of the related plants chain, not limited to the status of a certain device. Thus, the health condition of product line needs to be synthetically considered. In the second level, the input is the agent condition which is the coarse grain status in the first level, and the output is the product line condition.

The hierarchical ontology model of CASOA is presented in Fig. 4. The real-time status data as the fine grain status is updated to the data property of machining agent 1. The reasoning engine reasons out the coarse grain status of the machine model. And the coarse grain status is mapped to the data property of the product line model. After the second-round reasoning, the health condition of the product line is inferred and the system uses the product line condition to produce the correct responses to the order. When a new agent is engaged, the product line model will be updated through instantiating the new agent class.

C. AGGREGATION OF BASIC AGENTS

This section demonstrates the behaviors of agents by structuring their interaction in three scenarios. In these scenarios, agents comprehend the information from other agents and responds to them based on their knowledge base.

As Fig. 5(a) shows, when a customer submits an order, the suggestion agent should answer whether the product can be made. So, the suggestion agent inquires the machining agent whether the process can be completed. After confirming the process can be completed, the machining agent inquires the conveying agent whether the product can be delivered to the destination, and the conveying agent informs its status to the machining agent. Then, the confirmation message is uploaded to confirm the order. Finally, the suggestion agent creates an order task for the product agents.

Figure 5(b) shows how agents cooperate to execute the accepted order task. First, the customer places an order, and then suggestion agent creates an order task for product agents. The product agents disaggregate the task into several processes and publish them to the machining agents. The machining agents negotiate to choose the target machining agent for processing the product. Then, the target machining agent publishes its location to the conveying agents. The conveying agents negotiate to build a route to transport the product agents to the destination. After completing the

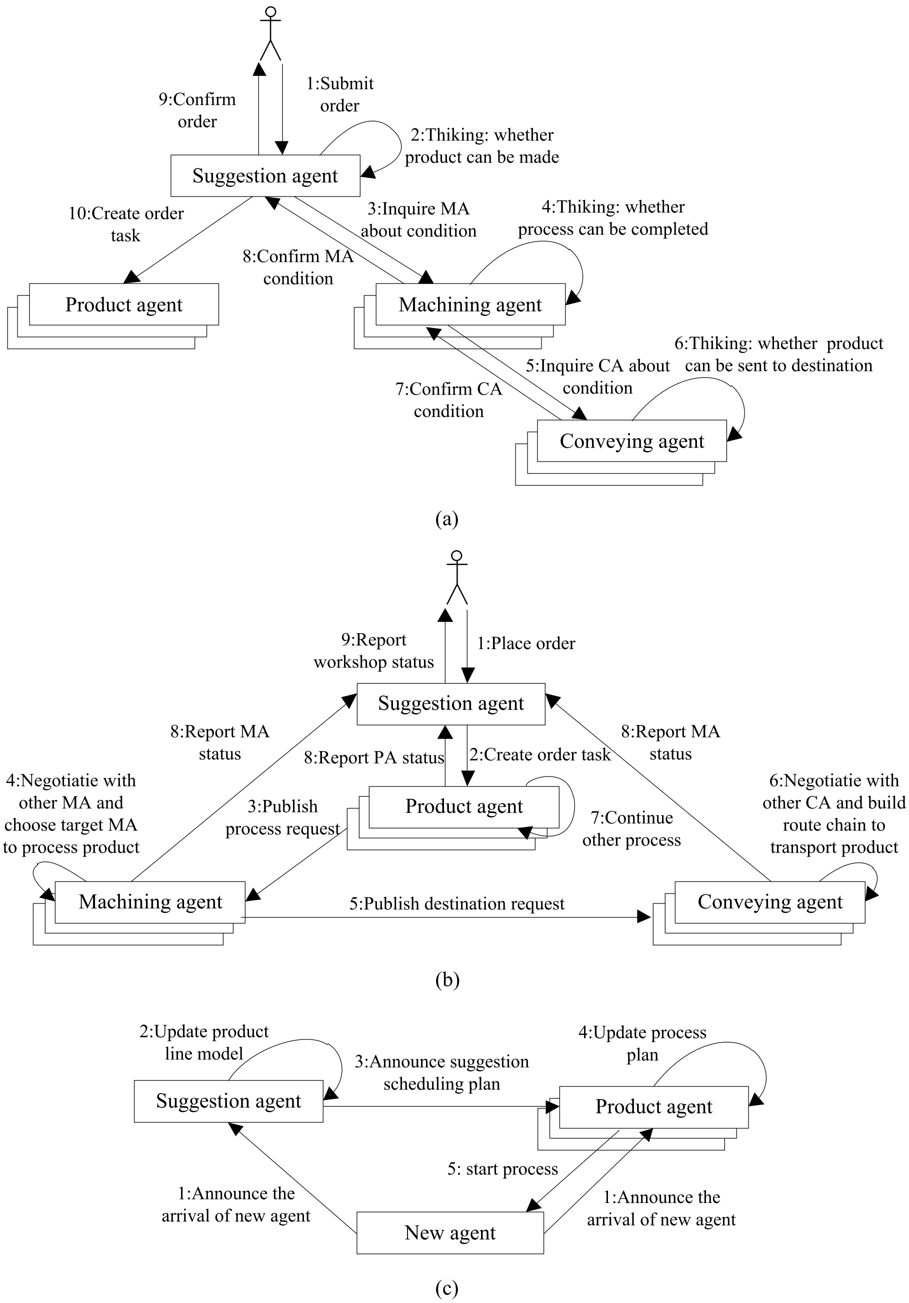


FIGURE 5. **Three scenarios of basic agents interaction behaviors. (a) The scenario of order receiving. (b) The scenario of basic agents' cooperation for executing the order task. (c) The scenario of adding an agent without shutting down the product line.**

process, the product agents continue to the next process until the task is finished. During this period, machining agents and conveying agents send their status to the suggestion agent. The suggestion agent analyzes the status data from lower resource in order to track and monitor products quality.

As Fig. 5(c) demonstrates, when a plant as a machining agent is dynamically added to the system without shutting down the product line, the new agent announces its arrival to the suggestion agent and the product agents which are under processing. After receiving the message, the suggestion agent updates the product line model to adapt a new manufacturing environment. Then, the suggestion agent produces new scheduling plans for the product agents. When the product agents receive the plans, they update the original process plans and request the new agent to initiate the new work.

III. COMMUNICATION METHODS IN CASOA

The interaction behaviors among agents in Section 2.3 are structured in a hierarchy through the effective and feasible communication mechanisms. The interoperability in smart

TABLE 2. Three communication methods in application.

Application	Real-time demand	Semantic demand	Communication method
Machine-to-cloud	★	★★★	OPC-UA
Machine-to-machine	★★	★★	DDS

manufacturing is divided into two layers: machine-to-cloud layer and machine-to-machine layer. Therefore, two communication methods, OPC Unified Architecture (OPC-UA), and Data Distribution Service (DDS) are employed respectively in these aforementioned layers to meet various real-time and semantic demands.

As TABLE 2 shows, in the machine-to-cloud layer, the cloud needs to be compatible with all data uploaded from the lower resources, so the semantic demand is the highest. In the meantime, it is unnecessary that the cloud provides real-time feed-backs to the lower resources to meet the real-time demand, since the resource scheduling is distributed and dynamic. The OPC-UA can build the customizable information model, so it offers high semantic transparency that suits machine-to-cloud layer.

In the machine-to-machine layer, due to the negotiation mechanism among agents, the real-time capability of communication is very important. The DDS purely relies on its data center that distributes the data, so it can meet the real-time demand in machine-to-machine layer.

A. MONITORING AND DATA ACQUISITION BASED ON OPC-UA

In the manufacturing process, plants are not only the production tools, but also the nodes of the information network. After data acquisition and analysis, the results can be applied to improve quality and efficiency of the production. The OPC-UA owns the information model and interfaces which support technologies such as ERP and Product Lifecycle Management (PLM). Thus, OPC-UA is used as a network bridge for monitoring and data acquisition in the scope of smart factory.

The presentation of the information model is similar to the ontology, thus, the information model can map to the agent ontology model and product line ontology model for achieving the interoperability between machine and cloud.

B. COMMUNICATIONS BETWEEN AGENTS BASED ON DDS

In the smart factory, if the data acquisition can be considered as the vertical information integration, then the communication between machines is the horizontal information integration. The DDS as a communication middleware supports publisher/subscriber mode which coincides with negotiation mechanism in MAS. As Fig. 6 shows, the agents subscribe the task topics which can be carried, and the product agent publishes the task. All machining agents obtain the task and reason out the current status to decide whether or not to bid for the task. After receiving the task intention, the product agent uses the knowledge base to choose the most available device. Finally, the product agent awards the chosen machining agent to complete the task. This kind of communication method based on topic data eliminates the information coupling of nodes and achieves the interoperability between agents in the lower resource layer.

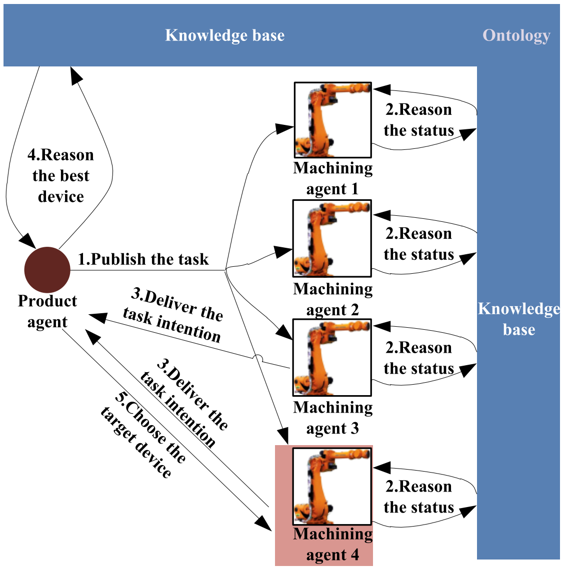


FIGURE 6. The negotiation among agents based on publisher/subscriber communication mode.

TABLE 3. The process capability of four machining agents.

Machining agent index	Process index and process time (s)			
1#	P_{001}(10)	P_{002}(5)	P_{003}(8)	P_{004}(5)
2#	P_{005}(6)	P_{006}(4)	P_{007}(7)	
3#	P_{001}(13)	P_{004}(10)	P_{008}(4)	P_{009}(5)
4#	P_{010}(3)	P_{011}(7)	P_{012}(5)	
5#	P_{011}(5)	P_{012}(9)	P_{013}(2)	P_{014}(3)

IV. EXPERIMENT

The experiment was conducted by using several plants and network devices in order to verify the proposed architecture with respect to its associated dynamic scheduling method. Five servers were used as a cloud to collect and analyze the field data. A group of machines including five robots as machining agents (MAs), and five PLC controlled conveyors as conveying agents (CAs) were interconnected to the cloud through Ethernet links. The objects are shown in Fig. 7.

A. EXPERIMENTAL SETUP

As shown in TABLE 3, five machining agents can complete the processes from P_{001} to P_{014}. Four types of products {A, B, C, D} were made and the total number of every type of product was 25 respectively. It was assumed that the workpiece type of entering product line is random and that interval time of adjacent workpieces ΔT obeys normal distribution.

Traditional static scheduling uses pre-planned method to allocate the jobs to the specified machines. It is usually

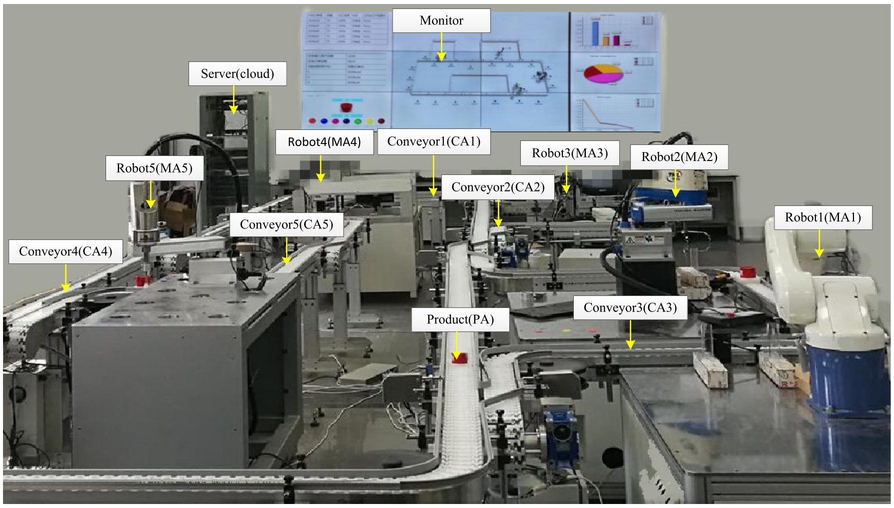


FIGURE 7. System layout in experiment [2].

TABLE 4. The traditional scheduling for four types of product agents.

Product type	Machining agent index and process time (s)			
A	1#(10)	4#(7)	2#(4)	
B	1#(8)	3#(4)	5#(9)	2#(7)
C	3#(4)	5#(5)	2#(6)	
D	3#(5)	1#(5)	4#(7)	2#(4)

based on the assumption of constant rate of resource supply known at design time. The pre-planned method represents that the centralized scheduler makes the optimal scheduling plan before the task executions and the plan cannot be changed in the process. The process plans of the pre-planned method for four types of product agents were as shown in TABLE 4. Three experiments were performed to verify the performance of product line: 1) the comparison of product line performances between pre-planned method and proposed method; 2) the comparison of product line performances for different configurations; and, 3) the comparison of product line performances before and after fault occurrence. In the experiments, the job delay ratio Z_j and machine utilization ratio Z_t were considered as performance indicators of the system, and they were defined by:

$$\begin{aligned} Z_j &= J_t \div J_{total} \\ Z_t &= T_t \div T_{total} \end{aligned} \tag{1}$$

where J_t is the number of delayed jobs, J_{total} is the total number of jobs, T_t is the actual processing time of the machine, and T_{total} is the total processing time. In the following, 1# denotes the machining agent 1, 2# denotes the machining agent 2, and so on.

B. EXPERIMENTAL RESULTS AND DISCUSSION

The results obtained in three mentioned experiments are presented in Fig. 8, wherein Fig. 8(a) presents the result of experiment 1, Fig. 8(b) and Fig. 8(c) demonstrate the results of experiment 2, and Fig. 8(d) shows the result of experiment 3.

Fig. 8(a) indicates the relation between ΔT and Z_j, wherein it can be seen that with increase of ΔT, Z_j decreases. The reason is the fact that when the workpieces concentrate excessively and the process speed could not keep step with the number of workpieces, J_t increases. In this figure, when ΔT is 6 s, Z_j of proposed method declines to about 3%. However, Z_j of pre-planned method is still 5% when ΔT is 10 s. The horizontal difference between two curves is about 4 s, which means the capability of proposed method is nearly doubled over pre-planned method. Therefore, proposed method is more adaptive than pre-planned method for processing random mixed-flow orders.

When ΔT is stable, the resources need to be reconfigurable to fulfill the external requirement. In experiment 2, ΔT is stable at 2 s. As Fig. 8(c) shows, when the total number of machining agents is five, Z_t of 2# is about 90%, which is abnormally high. This means that 2# is the production bottleneck that constrains the s production efficiency. Therefore, a new machining agent, agent 6 (6#) was added to work with original agents without shutting down the product line. Afterwards, the curve related to 6-machining-agents case shows that Z_t of 2# apparently declines. In addition, the curve related to 6-machining-agent case is smoother than that related to 5-machining-agent case. It means the machine utilization ratio becomes more balanced. As Fig. 8(b) shows, when the number of machining agents increases from five to six, the curve moves forward for 3 s. This reveals that when production bottleneck of a machining agent occurs, a bottleneck machine should be dynamically added to decline the process delay and to improve the production efficiency.

The occurrence of the device fault is unavoidable in actual production process. In experiment 3, ΔT was stable at 5 s. Namely, when the fifth workpiece entered the product line,

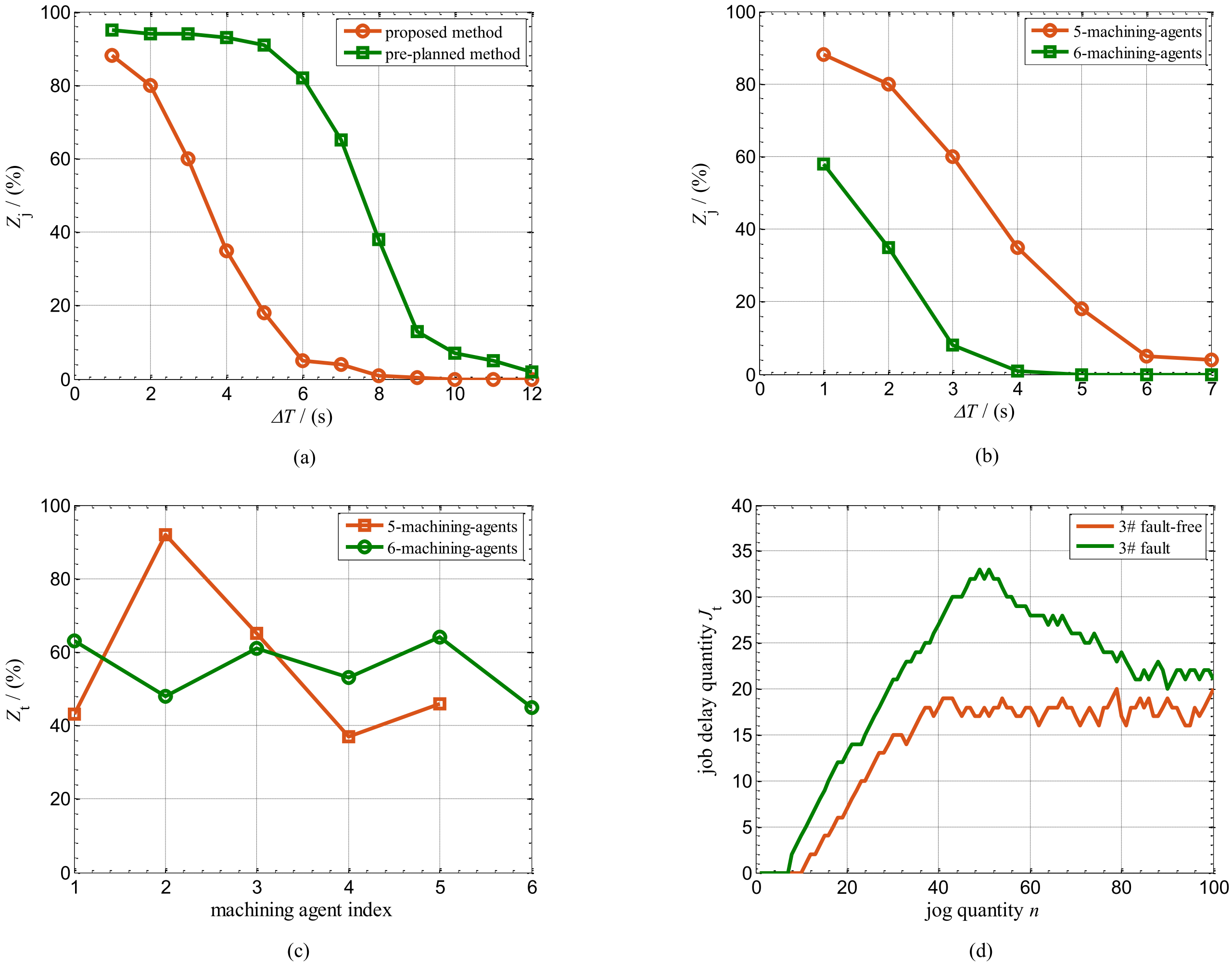


FIGURE 8. Experimental results. (a) The comparison of Z_j between pre-planned method and proposed method. (b) The comparison of Z_j for different system configuration. (c) The comparison of Z_t for different system configurations. (d) The comparison of Z_j before and after fault occurrence.

3# broke down and consequently could not accept the task. Before the fiftieth workpiece entered the product line, 3# was repaired. Figure 8(d) demonstrates that when fault occurs and n is greater than 5, Z_j rises sharply, but when n is greater than 50, Z_j drops slowly and approaches to the fault-free curve. This means that the proposed method based on MAS shows strong robustness.

V. CONCLUSION

In this paper, we present a cloud-assisted self-organized architecture called CASOA for agent-based manufacturing system. CASOA consists of four types of basic agents: suggestion agents, product agents, machining agents and conveying agents. Every type of agent concentrates on different functions of manufacturing system. The agents use the relevant communication methods to exchange their reasoning information based on their knowledge base. In addition, a cloud-assistant mechanism is introduced to coordinate the agents which are limited to local convergence.

The experimental results demonstrated and validated efficiency and reliability of the proposed architecture. The results showed that the proposed dynamic scheduling method has distinct advantages compared to the traditional static scheduling method. The proposed method shows high robustness and adaption to frequent product changes and disturbances.

REFERENCES

[1] J. Wan, S. Tang, Z. Shu, D. Li, and S. Wang, "Software-defined industrial Internet of Things in the context of industry 4.0," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7373–7380, Oct. 2016.

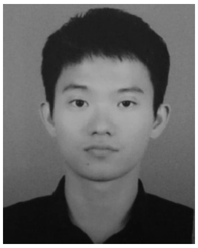
[2] D. Li, H. Tang, S. Wang, and C. Liu, "A big data enabled load-balancing control for smart manufacturing of industry 4.0," *Cluster Comput.*, vol. 20, no. 2, pp. 1855–1864, 2017.

[3] 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, doi: 10.1007/s11227-016-1879-4.

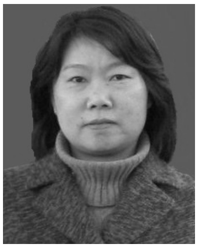
[4] J. Wan, S. Tang, D. Li, S. Wang, and C. Liu, "A manufacturing big data solution for active preventive maintenance," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.

[5] H. Seitz and G. Licht, "The impact of public infrastructure capital on regional manufacturing production cost," *Regional Stud.*, vol. 29, no. 3, pp. 231–240, 2016.

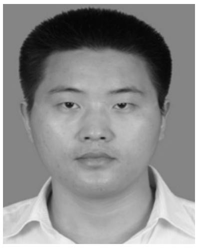
[6] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, to be published, doi: 10.1109/ACCESS.2017.2731945.
[7] D. Gyulai, B. Kádár, and L. Monostori, "Capacity planning and resource allocation in assembly systems consisting of dedicated and reconfigurable lines," *Procedia CIRP*, vol. 25, pp. 185–191, Jan. 2014.
[8] P. Renna, "Capacity reconfiguration management in reconfigurable manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 46, nos. 1–4, pp. 395–404, 2010.
[9] I. Niroomand, O. Kuzgunkaya, and A. A. Bulgak, "Impact of reconfiguration characteristics for capacity investment strategies in manufacturing systems," *Int. J. Prod. Econ.*, vol. 139, no. 1, pp. 288–301, 2012.
[10] D. Gyulai, B. Kádár, A. Kovács, and L. Monostori, "Capacity management for assembly systems with dedicated and reconfigurable resources," *CIRP Ann. Manuf. Technol.*, vol. 63, no. 1, pp. 457–460, 2014.
[11] M. Pirani, A. Bonci, and S. Longhi, "A scalable production efficiency tool for the robotic cloud in the fractal factory," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. IECON*, Oct. 2016, pp. 6847–6852.
[12] Y. Kristianto, A. Gunasekaran, and J. Jiao, "Logical reconfiguration of reconfigurable manufacturing systems with stream of variations modelling: A stochastic two-stage programming and shortest path model," *Int. J. Prod. Res.*, vol. 52, no. 5, pp. 1401–1418, 2014.
[13] R. Frei, N. Pereira, J. Belo, J. Barate, and G. Di Marzo Serugendo, "Self-awareness in evolvable assembly systems," *Thrombosis Haemostasis*, vol. 6, no. 3, pp. 355–382, 2010.
[14] P. Valckenaers and H. Van Brussel, "Holonic manufacturing execution systems," *CIRP Ann. Manuf. Technol.*, vol. 54, no. 1, pp. 427–432, 2005.
[15] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Comput. Ind.*, vol. 37, no. 3, pp. 255–274, 1998.
[16] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, Jan. 2015.
[17] A. L. Helleno, C. A. Pimentel, R. Ferro, P. F. Santos, M. C. Oliveira, and A. T. Simon, "Integrating value stream mapping and discrete events simulation as decision making tools in operation management," *Int. J. Adv. Manuf. Technol.*, vol. 80, nos. 5–8, pp. 1059–1066, 2015.
[18] B. Djelloul, M. Mimoun, and B. S. Mohamed, "Ontology based Web application reverse-engineering approach," *Infocomp J. Comput. Sci.*, vol. 6, no. 1, pp. 37–46, 2017.
[19] J. C. Nardi *et al.*, "Towards a commitment-based reference ontology for services," *Inf. Syst.*, vol. 54, pp. 263–288, Dec. 2015.
[20] E. Vysniauskas and L. Nemuraite, "Transforming ontology representation from OWL to relational database," *Inf. Technol. Control*, vol. 35, no. 3, pp. 333–343, 2015.
[21] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, and J. Lioret, "Context-aware cloud robotics for material handling in cognitive industrial Internet of Things," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2017.2728722.
[22] S. Wang, P. Zhang, and Y. Fan, "Centralized event-triggered control of multi-agent systems with dynamic triggering mechanisms," in *Proc. IEEE Control Decision Conf.*, May 2015, pp. 2183–2187.
[23] D. Zhang, Z. He, Y. Qian, J. Wan, D. Li, and S. Zhao, "Revisiting unknown RFID tag identification in large-scale Internet of Things," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 24–29, Oct. 2016.



HAO TANG was born in Hainan Province, China. He received the B.E. degree in mechanical engineering from Sichuan University, China, in 2014. He is currently pursuing the Ph.D. degree with the South China University of Technology. His research interests include intelligent manufacturing, industrial big data, embedded system, and motion control.



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. Thus so far, she has authored or co-authored over 180 scientific papers. Her research interests include embedded systems, computer vision, and cyber-physical systems.



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

ZHIJIE DONG was born in Zhaoqing, China, in 1982. He received the Ph.D. degree in mechanic and electronic engineering from the South China University of Technology, Guangzhou, China, in 2016. He has involved in image processing for many years. His current research interest is in inspecting technology using machine vision.