Article

A Novel Multi-Agent-Based Collaborative Virtual Manufacturing Environment Integrated with Edge Computing Technique

Xiaohui Zhang [1], Shufeng Tang [2,*], Xinhua Liu [1], Reza Malekian [3] and Zhixiong Li [4,*]

[1] School of Mechanical and Electrical Engineering, China University of Mining & Technology, Xuzhou 221116, China
[2] School of Mechanical Engineering, Inner Mongolia University of Technology, Hohhot 010051, China
[3] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa
[4] School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong NSW 2522, Australia
* Correspondence: tangshufeng@imut.edu.cn (S.T.); zhixiong_li@uow.edu.au (Z.L.); Tel.: +61-405-840751 (Z.L.); Fax: +61-703025 (Z.L.)

check for updates

Abstract: This paper proposes a multi-agent-based collaborative virtual manufacturing environment (VME) to save energy consumption and improve efficiency in the manufacturing process. In order to achieve the high autonomy of the manufacturing system, a multi-agent system (MAS) is designed to build a collaborative VME. In this new VME environment, edge computing is embedded to strengthen the cyber resource utilization and system economy. Moreover, an efficient communication channel between networks is proposed. The subsequent cooperation and collaboration protocols among agents are designed to ensure flexible and process-oriented operations. Furthermore, the fuzzy resolution algorithm is employed to resolve the competition conflicts among function-similar MASs in the distributed manufacturing scenario. Lastly, a simulation and case study are performed to evaluate the performance of the proposed VME in Internet of Things (IoT)-based manufacturing. The analysis results have demonstrated the feasibility and effectiveness of the proposed VME system.

Keywords: energy saving and efficiency; virtual manufacturing environment; edge computing; multi-agent system; internet of things

1. Introduction

The virtual manufacturing environment (VME) is an environment that executes manufacturing processes [1]. This environment enables the manufacturer to evaluate the production activities beforehand without using physical plants and facilities. With the rapid development of artificial intelligence, VME is often chosen as a platform on which work is carried out for various applications of intelligent systems, e.g., management systems, control systems, networks, and computing modes [2–4]. In a real manufacturing environment, an entire production procedure is comprised of subtasks in different function domains, such as process control, quality assurance, maintenance, and logistics. Therefore, collaboration among the fragmented domains is necessary to cope with both the internal and external fluctuating factors [5], e.g., energy cost-down plan, voice of customers, intense global competition, and shorter time-to-market strategy. Concerning the complexity of object management and collaboration, VME has to be developed in a modular and distributed way, and the distributed module requires a low coupling and high cohesion performance with strong independency principle [6]. A multi-agent system (MAS) is a loosely coupled agent network that solves problems exceeding single

agent capability or knowledge through interaction [7], and has proven to have great advantages regarding the construction of distributed systems, resources allocation, and adaptive reactivity. The goal of a MAS is to decompose large complex systems (software and hardware systems) into small, interoperable, and easy-to-manage systems. The merits of taking MAS into application embrace intelligent decision-making, high-level autonomy, interoperation with heterogeneous systems, high-level interactivity, etc.

On the other hand, the Internet of Things (IoT) has attracted great attention in the last decade. Many researchers have tried to construct IoT systems combined with thorough data manipulation and resource sharing mechanisms with the help of VME. For instance, large-scale IoT systems based on cloud computing have been prevailing in system establishment and management [8], in which cloud computing enables powerfully centralized data processing and storage with various service-provisioning paradigms such as "Software as a service" (SaaS). However, related difficulties that originated from the fixed features of cloud computing have included transmission latency, explosive bandwidth requests, and a high risk of privacy concerns. To solve these rigid dilemmas, embedding the edge computing into cloud computing is gradually becoming a development trend.

Bearing the above observations in mind, this article intends to carry on the comprehensive development of a collaborative VME through the integration of MAS and edge computing techniques with a case study. Distinguished from other research hotspots such as service, product life cycles, or supply chains, this study extends the study into production-oriented manufacturing, and concentrates more on daily activities.

The remainder parts are organized as follows: work related to VME, MAS, and edge computing are outlined in Section 2. Section 3 presents the system architecture of the developed VME with the elaboration of the MAS and edge computing framework. In Section 4, the system communication channel, the cooperation protocol, and the collaboration protocol in MAS are proposed. A fuzzy comprehensive evaluation method is adopted for the competition resolution. Section 5 demonstrates an exemplary network simulation, a case study based on edge computing platform, and a resolution of MAS competition. In the last section, conclusions and future research are generalized.

2. Literature Review

Recent publications relevant to this paper are mainly concerned with three research streams: VME, MAS, and edge computing. In this section, the correlative literature is listed and concluded.

2.1. Application of Virtual Manufacturing Environment

The field of VME has gradually broadened with the help of computing power. Various solutions for VME have been developed to overcome engineering deficiencies effectively, e.g., innovative design, the evaluation of production activities, the elimination of inappropriate results, and the maintenance of models in daily operation to achieve high-quality simulation [9]. Matsas et al. [10] built up a highly interactive fiber VME as an experiment platform where robots cooperate to switch and remove adhesive strips and fabrics from molds. Cai et al. [11] utilized an intelligent efficient autonomous hand behavior construction method for VME to improve the current mainstream hand animation method that totally depends on capturing hand movements. In the past decade, virtual reality technology has been prevailing in constructing VME. For instance, Vosniakos et al. [12] elaborated the approach that constructs a simulation model for robotic production units from scratch under virtual reality exemplification. Abidi et al. [13] embedded a lean simulation of a production process based on virtual reality technology in the virtual environment, which provided users with the interaction with a simulation model in three-dimensional (3D) format and in real time. Colledani et al. [14] developed a data-sharing model-based virtual factory environment that was exploited to afford a universal language for data exchange within all applications. Apart from the preceding techniques, emerging technology was also adopted to address stubborn problems in specific areas such as information-sharing barriers among different enterprises. Hao et al. [15] developed a messaging and data exchange platform that can impulse the interoperability and integration within

virtual factories partners. The above literature review has shown that VME can be extended in many application fields with high efficiencies and performances. However, VME has rarely applied for IoT-based manufacturing. The merit of VME motivates us to embed the innovative conceptual framework of smart manufacturing within it.

2.2. Multi-Agent System (MAS) in Engineering Application

MAS provides a particular intelligent aspect to resolve the coordination and cooperation between the product design, manufacturing, and even the whole life cycle of the product. It affords a more effective means for system integration, concurrent design, and intelligent manufacturing [16]. In the review, the multi-agent application in the manufacturing fields is focused. Cristalli et al. [17] introduced a new method to design a distributed production control system and demonstrated how to apply the multi-agent mechanism of a distributed system to enhance the integration of quality control in production processes. Feng et al. [18] designed a MAS formed by four types of agents, covering three layers: enterprise resource planning (ERP), a manufacturing execution system (MES), and the factory automation layer. In [19], a proposal for an IoT system represented by the user-orientated and environment-oriented ability was discussed. A system configuration platform furnished with a multi-agent framework was adopted to address the common problems of edge computing and traditional IoT architecture. Barenji et al. [20] introduced a novel architecture deploying an RFID (radio frequency identification) distributed monitoring and control system through multiple agents. The system is responsible for implementing different tasks and collaborating to enhance the flexibility and re-configurability of the production system. Marks et al. [21] utilized MAS to respond to manufacturing alternation requests and create adaptive choices for the equipment based on the system's mechatronic model. Besides, Durica et al. [22] presented MAS and adopted a novel bio-inspired algorithm in energy-efficient manufacturing. García-Magariño et al. [23] proposed a tool named MTBE (model transformation by example) for the creation of model transformations in MAS that are based on the generation of model transformations as an example. It overcomes the limitations of other similar tools in the sense that it can generate many-to-many transformation rules. As reviewed, in short, with the help of the high autonomy of MAS, the management of manufacturing can be greatly simplified. One contribution of this paper is to design MAS to realize the collaborative manufacturing environment.

2.3. Edge Computing

Even though the public cloud service mode provides a relatively secure and convenient environment for enterprises as an alternative, transmission latency and congestion remain to be considered as the rigid characters for cloud computing. On the other hand, due to cost and efficiency factors, minor enterprises still want to have a solution that makes the most use of their local computing devices [24]. A flexible technology is needed to take up such challenges. Edge computing, as an emerging distributed computing technology and an improvement of cloud computing, is launched. Compared with cloud computing, there are the following advantages by applying edge computing:

(1)　Not all the data is necessary to be aggregated to the backstage data center (DC) such as OpenStack for centralized processing. Edge computing can perform the intelligence at the network edge that is close to data source or sensor to obtain rapid local access and load distribution [25]. This specific merit will decrease communication delays as well as the entire size of the data to be migrated to the remote end through public and private DCs.

(2)　Data processing performed by edge computing enables the application of a cluster of mechanisms to implement the analysis of acquired signal and instant decision making on-site [26].

(3)　Edge computing can utilize idle resources effectively, which helps further reduce the load of central processing.

(4)　Edge computing enhances the offline protection of privacy and security [27]. Although a cloud-based solution may also help offload the computation overhead of security protection, users are required

to hand in their security keys to cloud servers. Edge computing preprocesses a matched algorithm on the acquisition processing; the data can be transmitted only after privacy protection, which means the security remains as security keys concealed by users.

Currently, the application of edge computing concentrates mostly on the mobile field and IoT. In [28], a new robust dynamic edge network architecture (RDNA) based on edge computing was proposed that leverages the advantages of recent advances in mobile devices to acquire left-off resources and mitigate the crowding of networks in a dynamic way. Amjad et al. [29] proposed a framework solution based on cognitive edge computing to achieve efficient distributed resources utilization that provided end users with a dynamic flexible extension of cloud computing and devices. Tran et al. [30] designed a context-aware, real-time cooperation structure that sits on the edge of a radio access network. The highlight of this article also lies in the integration of heterogeneous resources to the edge. Despite the merits of quick response and high computing efficiency, Hsu et al. [31] described a novel reconfigurable security architecture based on edge computing that uses an edge device near users, i.e., security agents, with the view of simplifying the key governance and offloading calculation costs for algorithms on IoT devices. As an emerging technique, edge computing can provide more flexible solutions than cloud computing, especially for SMEs (small and medium-sized enterprises). Therefore, one of the contributions in this paper is to apply this technique to IoT-based manufacturing, with the view of saving resources (e.g., energy and means of production) and increasing communication efficiency.

2.4. Discussion

Each aforementioned advanced technology or model has its own emphasis and plays an important role in the development of manufacturing industries. There are still some observations. (1) Most researches have focused on a single technology rather than comprehensiveness. However, to ensure numerous features of an IoT-based manufacturing system simultaneously, a combination of different technologies is essential. (2) As far as we know, few studies have integrated MAS together with edge computing technique into the VME. It has meaning to give play to their own advantages for supporting the mobility requirements and geographic independence of VME. Bearing the above observation in mind, an edge-based MAS architecture to realize the functionality of a collaborative VME is proposed. It is worth mentioning that although edge computing has many advantages, cloud computing remains in the proposed VME, as it executes more powerful central resource management.

## 3. Architecture of the Proposed Virtual Environment

To demonstrate the research work more clearly, an overview of the research process of the proposed study model is summarized. Firstly, the overview of the proposed VME with a hierarchical description of each working environment is presented, and the cooperation framework of the system layer in VME is specially illustrated in Section 3.1. The MAS architecture and individual agent functionality are depicted with an example in Section 3.2. The network communication and agent interactions in MAS will be elaborated in Section 4.

3.1. Overview of the VME

Figure 1 depicts an overview of the proposed VME. The VME covers three layers: a service layer, an edge-based system layer, and a user interaction layer. Next, each layer with their functionalities is elaborated separately.

The service layer attends to the cloud infrastructure. The cloud processing module in this layer takes charge of big data processing and analysis, while the storage module empowers an abstraction for flexible data storage. Since the IoT in the manufacturing field contains plenty of physical machines furnished with heterogeneous systems in the form of configurations, models, and semantics, it is inevitable to shield such heterogeneous systems to provide uniform standard service. Here, a virtual machine technique [32,33] is employed to resolve such problems. It maps the physical machines distributed in scattered geographical

locations to virtual machines on the cloud platform. All the resources are centralized, integrated, and further rented out on a pay-as-you-go model [34]. When any users intend to access the resources through the gust OS (operation system) under the SaaS paradigm, relative virtual machines will be triggered and organized availably for sharing through cloud connectivity. This cloud infrastructure brings the advantage of high scalability, because alternative service providers can be arbitrarily selected according to data types, e.g., structured/semi-structured binary and semantic data.
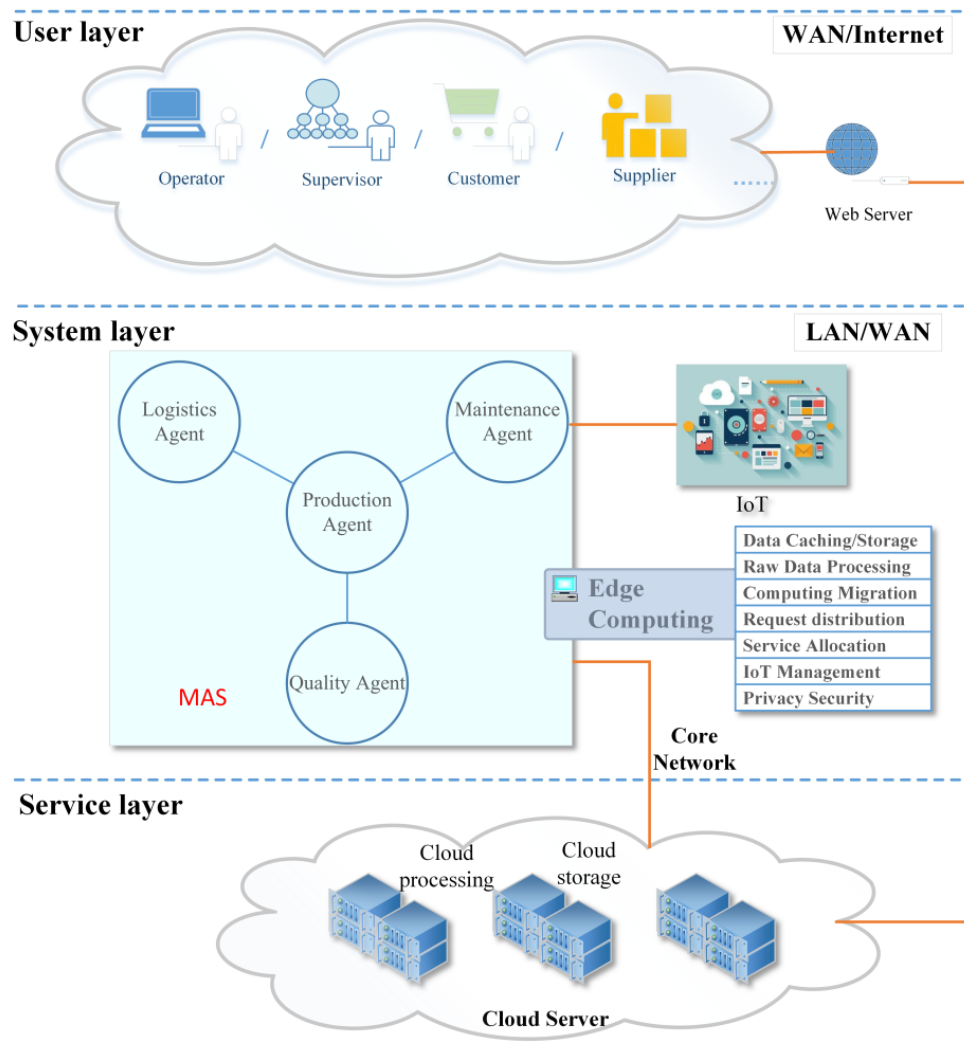
Figure 1. Overview of the virtual manufacturing environment (VME) architecture.

The edge-based system layer is deployed in relation to an IoT system and MAS architecture. Figure 2 illustrates the cooperation of the key components that are relevant to this layer. In the proposed framework, edge computing is granted permission to shrift the required computational power out of the cloud center, and locate it closely to end-user devices (e.g., test equipment). That is, edge devices acquire the equipment data and transfer it to the edge server (e.g., intelligent gateway) through network interconnection. An edge gateway facilitates the necessary performance of storage and computing resources, management of membership, and logic operation. On the other side, the allocation of edge computing is regulated by MAS. MAS proceeds reasoning and makes decisions that act on the end effectors (work site) based on information fusion. The edge server keeps delivering the aggregated data to the cloud platform for further analysis and storage.
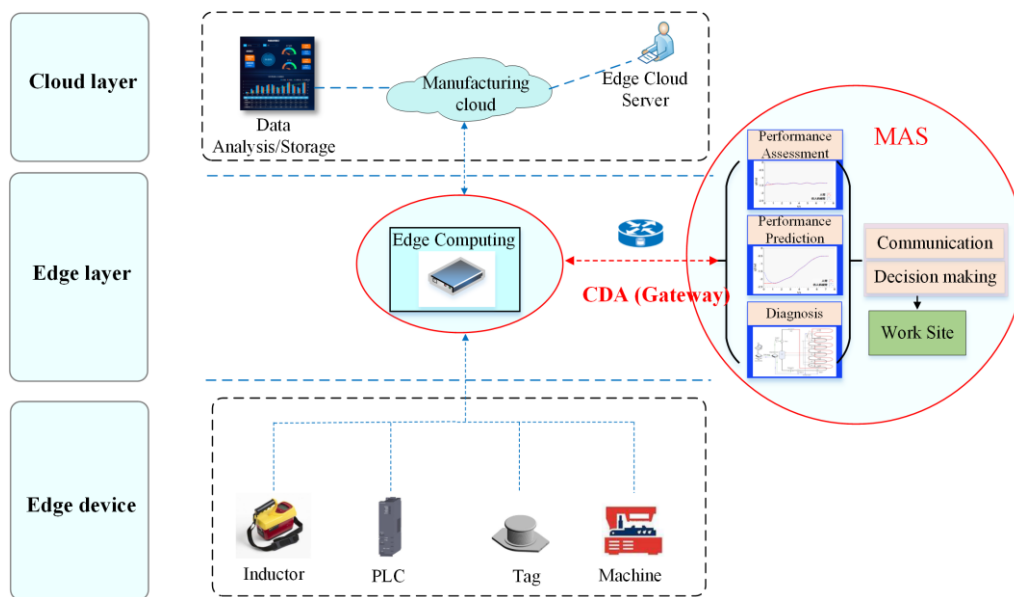
Figure 2. Cooperation framework of the system layer.

The user interaction layer generated by user abstraction comprehends service requests and preferences via human–computer interaction. As internal users, the supervisors and operators have access and permission to retrieve and invoke the underlying data with LAN (local area network) /WAN (wide area network); as external users, the customer and supplier can share their actual product, delivery, and quality information in a standard data format through cloud connectivity in XML. Since the communication and interaction are implemented on the centralized platform, the user interaction layer is virtually a cloud-based application layer.

3.2. Multi-Agent Architecture and Individual Agent Functionality

Figure 3 schematically demonstrates the system layer under the development of a MAS-based framework. The framework also specifies a domain-based hierarchical structure. The MAS is equipped with a set of intelligent agents (production agent, quality agent, logistics agent, and maintenance agent) that regulate their subagents in each respective domain. There are two hierarchies. In the first hierarchy, the quality agent, maintenance agent, and logistics agent collaborate with each other and are harmonized by the production agent for achieving a common production target. These four intelligent agents own the capabilities of data processing, aggregation, and storage, and can act as an edge server (intelligent gateway). With the features of intelligent gateways, the agents realize the protocol transformation between different types of perceptual networks and communication networks both in WAN and LAN. According to the categories in [35], and taking the functionality into account, they are designed as a kind of hybrid collaborative and deliberate agent (CDA) whose structure is presented in Figure 4 as an example. A CDA consists of database module, repository module, control module, and communication interface module. It can be abstracted using a four-tuple:
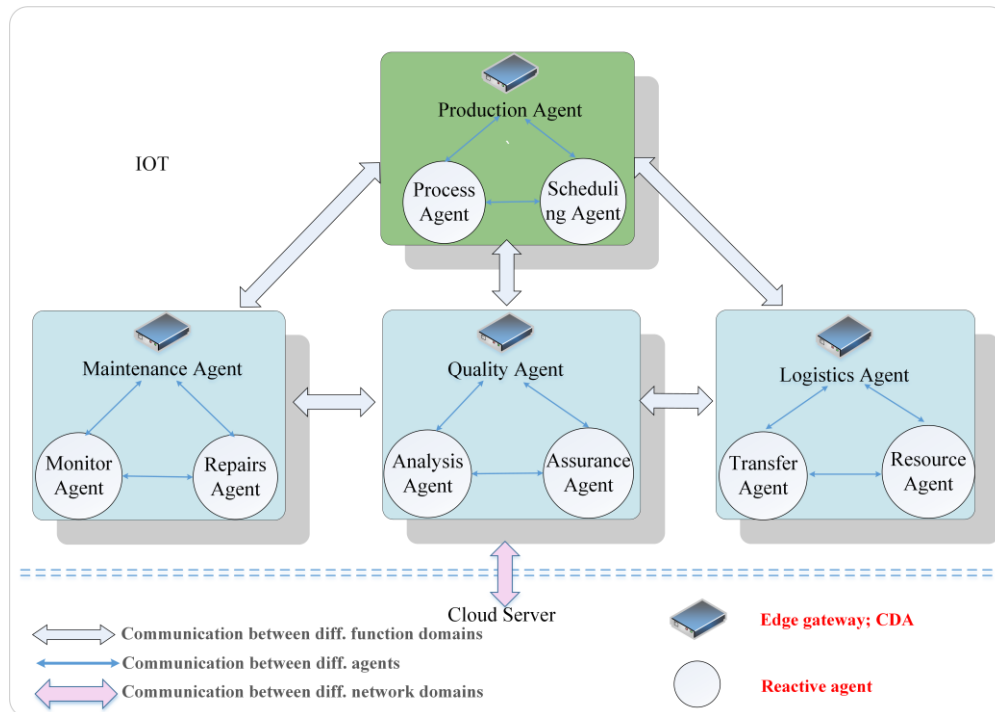
$$Agent = <D, R, C, I>.$$

where:

D: database. The database holds all the associated data for processing. It keeps the operation logs and records activities.

R: repository. The repository stores the knowledge that forms the base of decision-making, and supports activities.

C: control. The control module makes decisions and controls the activities. Functions such as cooperation with other agents and task allocation are executed by the control module.

I: communication interface. The communication interface provides access from agent to agent, the network, and the environment. It is namely a primary conduit for communication between agents and externality.


Figure 3. Schematic description of multi-agent architecture.
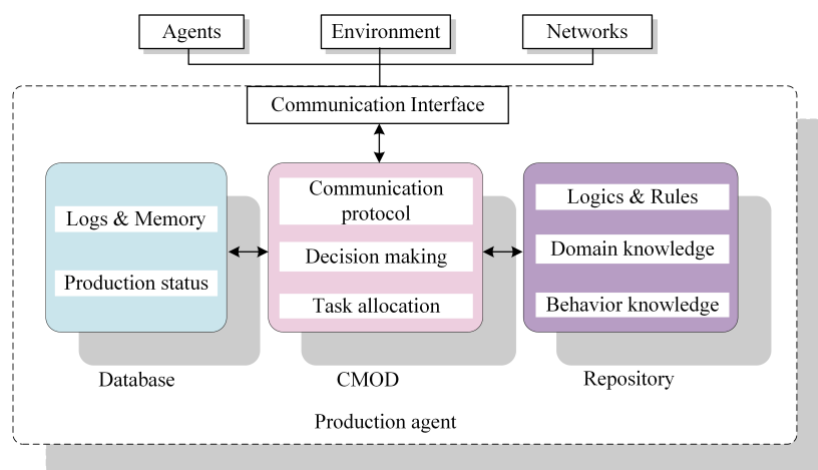
Figure 4. Exemplary structure of collaborative and deliberate agent (CDA).

The CDAs in the first hierarchy manage and dominate the performances of their subagents (e.g., monitor agent, resource agent, and scheduling agent in Figure 3) that stay on the second hierarchy of the multi-agent architecture. These subagents collaborate with each other to support the decision makers (CDAs) in their own functional domains. According to the work specification, they are designed as a type of reactive agent. From Figure 3, it can be seen that in each specific domain, one CDA harmonizes two reactive agents. Therefore, there are in total eight reactive agents in a MAS. The exemplary structure of the reactive agent is shown in Figure 5. It is worth mentioning that both CDAs and reactive agents possess certain abilities to resolve problems autonomously through continuously perceiving environmental information, updating their own information database, and communicating with each other. In addition,

each agent in the system layer is furnished with decision-making functions and expert knowledge in dedicated fields, which help observe different aspects, space, and characteristics of the manufacturing environment. The differences between CDA and reactive agent stay mainly on the decision-making scope and data-processing ability.
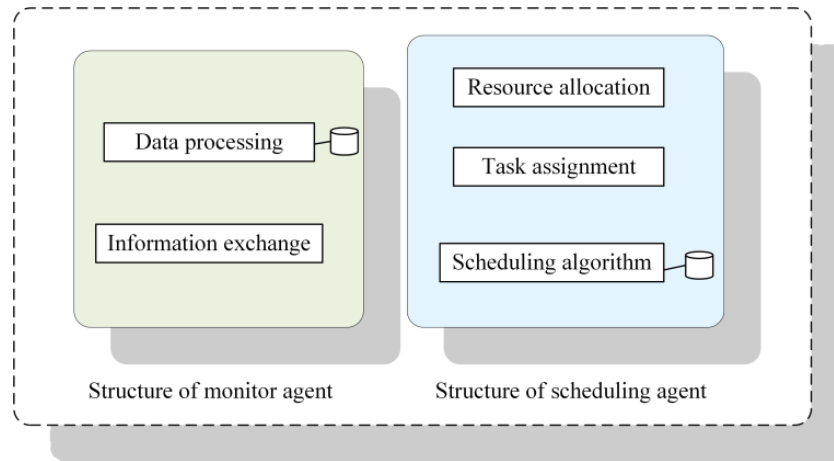
Figure 5. Exemplary structures of reactive agents.

Next, the functionalities of all the agents concerning each domain are elaborated as follows:

(1) Production domain: The production agent harmonizes other CDAs and collaborates the subagents in the production domain. The scheduling agent dispatches production tasks according to the actual overall production status dynamically. The process agent adjusts the production processes with respect to the product or material changes from the customer or supplier side.

(2) Quality domain: The quality agent takes the overall responsibility of quality assurance through cooperating with the two subagents. Work refers to the quality assurance carried out on-site and mainly includes a pretest, sampling inspection, and emergency measures. The quality agent acquires quality data along the production cycle. The analysis agent is in charge of the measuring output and the follow-up analysis. It gives a necessary quality warning when an abnormal event occurs. The assurance agent decides how to handle the nonconforming products based on the quality handling knowledge, e.g., blocking, reworking, scraping, or releasing.

(3) Maintenance domain: The maintenance agent has two subagents: a monitor agent and a repairs agent. The monitor agent executes the data acquisition from several sensors in the IoT environment. The status of equipment such as wear, tear, damage, collision, and other abnormality are detected through analyzing equipment data. Consequently, proactive maintenance and breakdown identification can be implemented. The repairs agent disposes of the repair and spare parts information, and interconnects with the maintainer on site. If a maintainer intends to change one spare part for a device, he will invoke the database in the repairs agent with the help of radio frequency identification (RFID) or another embedded platform to retrieve the available substitutes. The maintenance agent manages the performances of two subagents and assists the production agent around the lead-time.

(4) Logistics domain: The logistics agent consists of two subagents. The transfer agent supplies the resources such as purchased parts and self-making parts through routing inspection for achieving a dynamical resource allocation. Due to the distributed manufacturing locations and inhomogeneous tasks that originate from customer demand, artificial intelligent algorithms are employed by the transfer agent to obtain the best routing planning. The resource agent formulates reasonable inventory rules for all the resources that circuit between the logistics system, and takes the responsibility of connecting with suppliers and customers for buffer balance. Since modern

management is committed to promoting a zero-inventory strategy, the allocation of resources for plant areas is restricted and limited. Too little or too much buffer will create an inventory alert, and production activities will be greatly affected. Both the transfer agent and resource agent collaborate with each other as subagents under the guidance of the logistics agent.

4. Network Communication and Agent Interactions in MAS

Communication is a key element of system operation. The accuracy and efficiency of communication between entities determine the performance of the proposed system architecture. On the other hand, a VME needs to forge a platform where information could be exchanged freely by any task participants. In this section, the communication channel between networks and particular interaction (cooperation, collaboration, and competition) protocols for the MAS are proposed.

4.1. Network Communication Channel

As aforementioned in Figure 4, the CDA such as the production agent plays the role of an intelligent gateway that integrates the performance of protocol, application, and safety. One primary functionality of the gateway is to translate data using diverse communication protocols, data formats, or languages. Since the CDA in the proposed framework interconnects with the IoT and can access the global cloud server, the communication between networks can be characterized via edge gateways. The edge gateways are deployed in the edge network communication, which is shown in Figure 6.
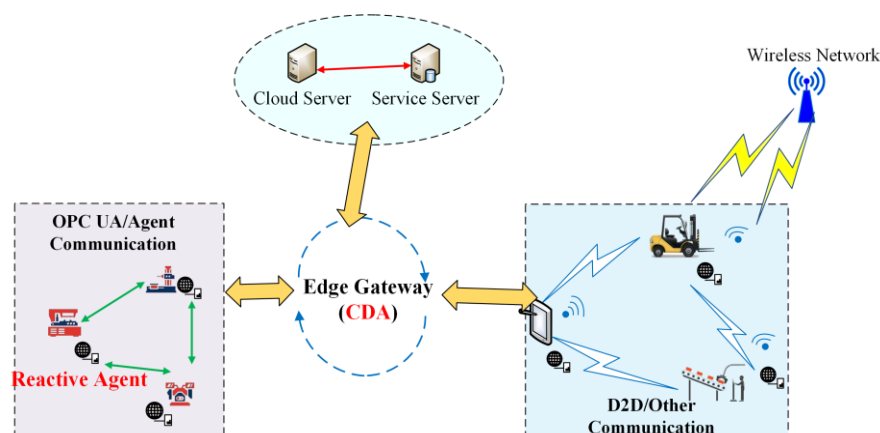

Figure 6. Schematic description of the network communication channel.

In Figure 6, the OPC UA OLE (object linking and embedding) for Process Control, Unified Architecture) [36] affords a complete, secure, and reliable cross-platform architecture for information interaction. It enables the system to reach the semantic interoperability as well as the illocal perception of the manufacturing. The server of the OPC UA could be constructed using the edge nodes. To deploy edge computing, a software container on the gateway can be installed. To ensure the requirements of low congestion and latency, edge computing can be regarded as a time-aware scheduler by introducing the time-sensitive networks (TSN) [24]. In such a scenario, edge computing allocates the information communication and affords the real-time capability on the time nodes.

At the device terminal, device-to-device (D2D) communication is a new technology that allows terminals to communicate directly by multiplexing cell resources under the control of the system [37]. D2D realizes a convenient resource allocation with the base station in the manufacturing environment. To some extent, it solves the problem of resource scarcity in wireless communication systems. With D2D, smart mobile devices can interact freely with each other through the wireless access. In the case of no network coverage, devices can separate their connections from nearby devices and form local sub-networks intelligently. Then, they can access the core network through the intermediate connections [24].

4.2. Agent Communication

Virtually, communication between function domains is the communication between agents. There are commonly three kinds of communication modes between individual agents: a message-based mode, an event-based mode, a signal-based mode. Here, the message-based mode for the illustration of agents' communication is chosen due to its intuitive feature. Under this mode, messages in MAS can be transmitted through an ontology model to achieve excellent compatibility and feasibility. Recently, ontology technology has been prevailing in the representation of a set of concepts and their relation in specific domains. Among most ontology languages such as the resource description framework (RDF), DARPA agent markup language + ontology interchange language (DMAL + OIL), extensible markup language (XML), simple HTML ontology extensions (SHOE), and web ontology Langauge (OWL), XML was verified to have great advantages regarding the readability and exchange of information. It is the semantic markup language that was designed as the vocabulary extension of the resource description framework (RDF), which provides more modeling primitive constructors and possesses clearer and more formal semantics [38]. In this study, XML is adopted as the ontology to describe the communication between individual agents. That is, different types and formats of messages will be transformed into XML representation through the semantic annotation firstly. The annotated message in the agent can be processed through the editor and explorer module. At last, an output message in a formatted document depicted by RDF is created, and other agents can recognize it. Figure 7 illustrates the XML and RDF-based message transmission procedure between two agents.
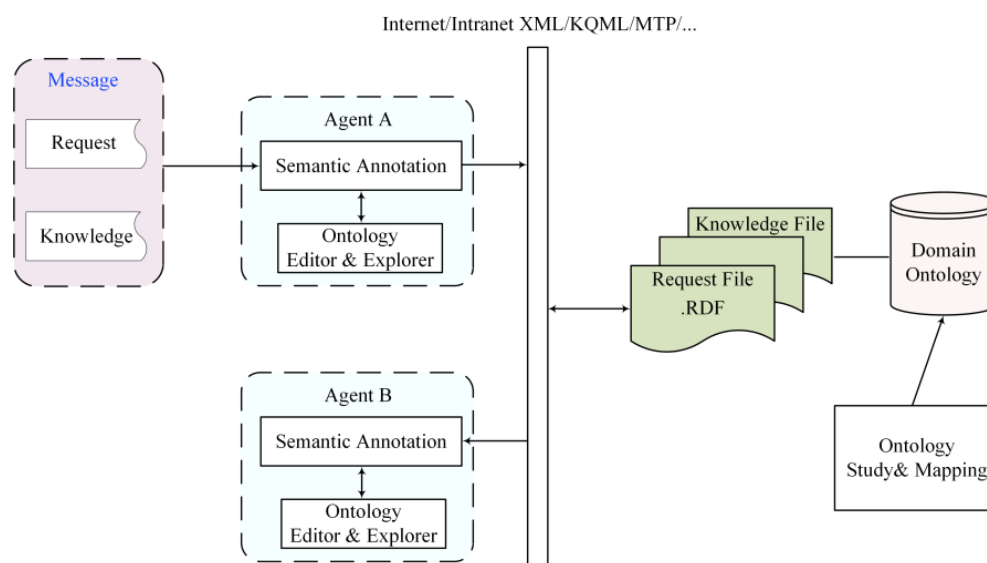
Figure 7. Agent communication mechanism within a multi-agent system (MAS).

4.3. Agent Cooperation and Collaboration

4.3.1. Agent Cooperation Protocol between Function Domains

Since one agent forms merely an inaccurate picture of the whole MAS, the system will be tangled without cooperation between the individuals. In general, the purpose of cooperation is to fulfill the targets of agents furthest. Complied with cooperation, the interactive agents need to adjust their targets and corresponding behavior, especially regarding the utilization of resources, because the behaviors of multi-agents interfere with each other by all means to achieve common objectives. The basic content of cooperation concludes target cooperation, resource allocation, procedure cooperation, and result cooperation. In this study, the cooperation is depicted in our hierarchical architecture based on organization structuring, which defines the responsibility, capability, interaction, and control flow. Figure 8 illustrates the cooperation protocol of the CDAs.
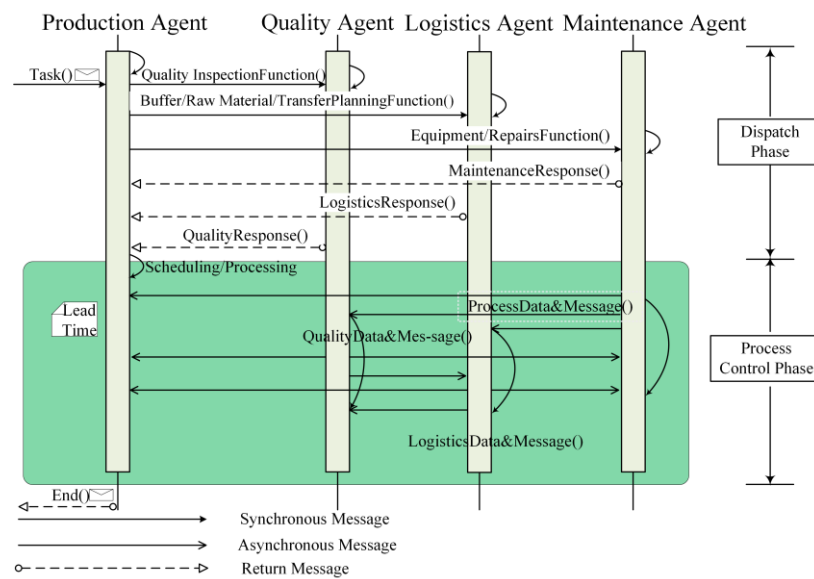
Figure 8. Cooperation protocol of the collaborative and deliberate agent (CDAs).

4.3.2. Agent Collaboration Protocol in Individual Function Domain

The relationship between individual agents in a dedicated function domain is mainly collaborative, such as that between the monitor agent and repairs agent in the maintenance domain (please see Figure 3). The collaboration behaviors within a function domain emphasize that the domain has a unifying overall goal and is generally applicable to situations where the targets and intentions of each agent are consistent. Multi-agent collaboration can extend the application fields and improve the performance of individual agents and the overall flexibility of the system. Figure 9 shows an example of collaboration protocol in the logistics domain. It is worth mentioning that "collaboration" distinguishes from "cooperation" through the action definition of individual agents. Even though the CDAs in the upper hierarchy are granted a specific role to manage the subagents, they are still harmonized by the production agent. Therefore, such CDAs participate both in "collaboration" and "cooperation" activities.
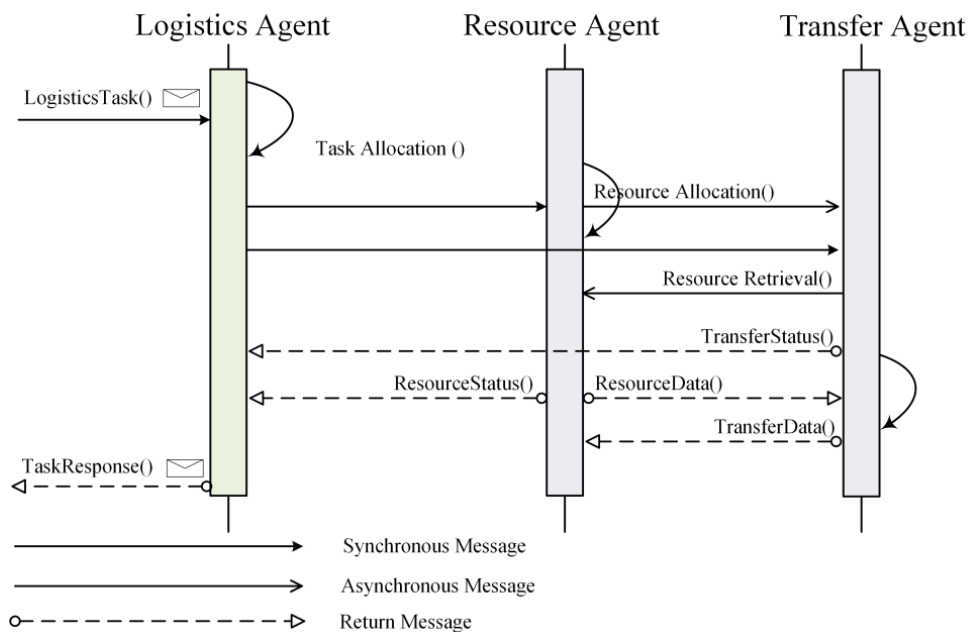

Figure 9. Collaboration protocol within logistics function domain.

4.3.3. Example of Agent Cooperation and Collaboration

An example is introduced to show how the MAS operates. After one routine sampling inspection, the quality agent delivers a report with quality related data as an attachment, which shows that one sampling from one machine of the production line is contingently out of tolerance. The production agent immediately starts an analysis using big data matching with the aid of a knowledge-based database and locks in a few potential factors (e.g., the machine and tool, the operator, the process, the environment, and the raw material) that might cause the quality problem. Then, the production agent will assign parallel tasks and collaborate with other CDAs: the quality agent will organize its subagents (analysis agent) to check whether the dimensions of the troubled batch raw material are qualified, and send the result back to production agent. The maintenance agent will organize its subagent to examine whether the machine geometry and the tools are in good condition, and send the feedback to the production agent. In the meantime, the report from the maintenance agent indicates that the bad conditions of machine geometry had caused the above problem. Then, the production agent will issue commands, and the machine will be halted for maintenance immediately. After the maintenance agent accomplishes the maintenance process, the adjusted samples will be transferred to the quality domain for dimension checking by the logistics agent. If all of the checking results from the different samples are stable regarding the tolerance and meet the standard, the production agent will organize the quality agent and maintenance agent to release the blocked parts and suspended the machine, respectively. Then, production resumes.

4.4. Competition Resolution Algorithm

Due to the distributed characteristics in intelligent manufacturing, competition among MASs with identical functionality is viewed always as a prominent factor for modern enterprises. For a collaborative manufacturing initiation system, it is vital to select the appropriate potential executors for the tasks. The evaluation of manufactural activity is the premise of making choices, and establishing a scientific evaluation index system is a good basis for evaluation.

The fuzzy comprehensive evaluation method (FCEM) [39] is a qualitative and quantitative system analysis method that has been widely extended in application fields of decision-making, prediction, and evaluation. In this study, FCEM is implemented as a tool for tackling the conflict problem caused by competition. The FCEM procedure is presented based on [40]:

(1) A production task is claimed. A set of bid MASs B with k elementsand a set of evaluation arbiters E with n elements are created.

(2) A set of evaluation criteria C concerning E is constructed as:

$$C = \{c_1, c_2, c_3, \ldots, c_m\} \tag{1}$$

where C_i specifies the key factors that impact the manufacturing system.

(3) The system builds up a set of weight coefficients W for the evaluation criteria as:

$$W = \{w_1, w_2, w_3, \ldots, w_m\} \tag{2}$$

(4) Each evaluation arbiter gives its estimation for a bid MAS according to the evaluation criteria C, and a fuzzy relation matrix R can be formulated as:

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix} \tag{3}$$

where r_{ij} specifies the evaluation score when the evaluation agent j ($j = 1, 2, 3, \ldots, n$) evaluates the bid MAS based on evaluation criteria i ($i = 1, 2, 3, \ldots, m$).

$$\sum r_{ij} = 1 \tag{4}$$

(5) The evaluation criteria can be based on quality, cost, and experience. So, an evaluation grade set is established by:

$$V = \{v_1, v_2, v_3, \ldots, v_n\} \tag{5}$$

where v_1—worse; v_2—bad; v_3—normal; v_4—good; and v_5—better.

(6) Establish a fuzzy set:

$$\underline{B} = (b_1, b_2, b_3, \ldots, b_n) \tag{6}$$

where b_j (j = 1, 2, 3, ... , n) reflects the status of comment v_j in the comprehensive evaluation, and $\underline{B}$ relies on the weight of each factor:

$$\underline{B} = R \cdot W \tag{7}$$

(7) Applying the fuzzy operator model ($\cdot$, $\vee$) expressed as:

$$b_j = \overset{m}{\underset{i=1}{\vee}} (r_{ij} w_i) = \max(r_{1j} w_1, r_{2j} w_2, \ldots, r_{mj} w_m) \tag{8}$$

The final evaluation V_i for the MAS i can be derived with the help of the maximum degree of the membership method.

(8) Repeat steps (4) to (7) to obtain the evaluation for all the MASs.

(9) Compare all the evaluations and choose the optimal one.

(10) End.

5. Case Studies

5.1. Case Study 1: Simulation of the Proposed VME

In this section, a network simulation software eNSP® (open source) is applied to construct a virtual production system containing 20 virtual machines as infrastructures, four gateways as edge computing actuators, one wireless access as cloud connectivity, and one PC terminal as the central processing unit. Table 1 lists the basic events that occur in manufacturing, where data packages and messages are transmitted through networks that interconnect the global system. The network transceivers are organized through triggering the events in intensive intervals. Recovery activities denote the event-handling procedures where multi-agents collaborate to resolve the predicament that occurred. The event-handling procedure consumes a huge amount of bandwidth due to the multirole coordination across different function domains. In the simulation, three architectures are tested. They are cloud architecture, edge architecture without MAS, and edge + MAS architecture. Figure 10 demonstrates the simulation results. The X and Y coordinates specify the time and average throughput of the networks, respectively. As can be seen in Figure 10, at the beginning of the production, the overall throughput proliferated until reaching the normal production state. This indicator is impacted after triggering the abnormal events. Especially when the network suffers from sampling inspection failure and machine breakdown issues asynchronously (overlap area), the throughput capacity of the two edge-based architectures exhibit better tenacities. The results show that the two edge architectures have overcome the cloud architecture in offloading data processing. Compared with cloud-based architecture, the average throughput has been improved by more than 50% when applying the edge architectures. This could be attributed to the distributed characteristics of edge computing. On the other hand, among two edge architectures, the proposed edge + MAS architecture has also shown superiority over the normal one. The phenomenon can be attributed to the efficient autonomy provided by the MAS.

Table 1. Basic events of the simulative manufacturing environment.

Event	Occurrence Scope	Information Trigger	Trigger Interval	Recovery Activity	Latency Affect
Routine production	per machine	machine	1 s	0 s	+
Sample inspection	per product	machine	2 s	0 s	+
Buffer balance	raw material	smartphone	5 s	2 s	+
Buffer balance	finished product	smartphone	5 s	2 s	+
Sample inspection fail	20% machines	machine	30 s	15 s	+
Breakdown	20% machines	machine	35 s	20 s	+
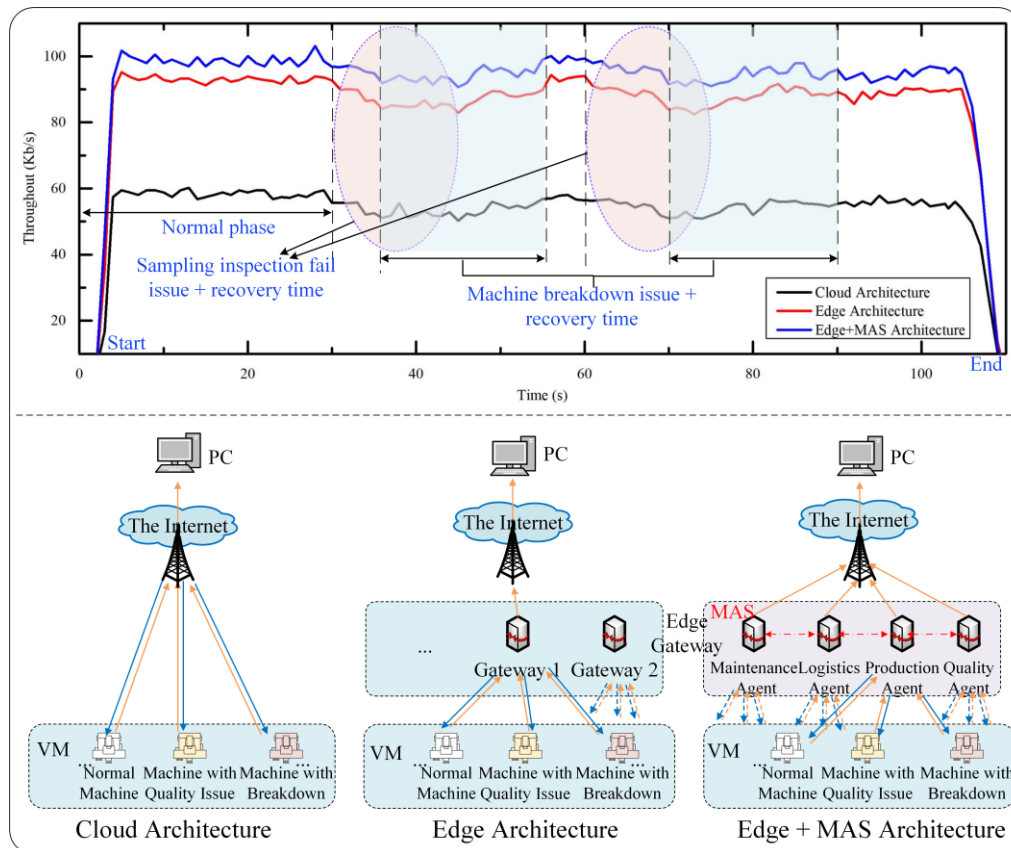
**Figure 10.** Simulation result of the virtual manufacturing environment (VME).

5.2. Case Study 2: Experiments of the Proposed Framework

In this section, a distributed manufacturing environment [24] is built using a Hadoop-based framework. The Hadoop Distributed File System (HDFS) is set to implement the real-time acquisition and analysis of the local data, which is composed of machine log data and status data. The data integration is loaded on a Raspberry Pi 3b+ with a Linux system. On the Raspberry Pi, an OPC UA server plays a role as monitor agent to transmit unintegrated information and pre-process the raw data from PLC (Siemens S7-300) of the experimental platform. A semantic model based on OPC-XML-DA (data access)standard is built to realize the data fusion of heterogeneous devices. Meanwhile, to achieve efficient information interaction, the function of MAS is implemented through Raspberry Pi.

In the laboratory, various customized boxes are to be stored in the warehouse of a smart production line. The customer service is carried out with help of a private cloud. To realize the communication with quality, an ad hoc network is established as an edge network near the device edge. In the experiment, the robots are deployed as edge nodes of the backbone network. They are also considered as the actuators of the production agent. The experimental platform is shown in Figure 11. As can be seen in

Figure 11, the tasks are allocated through a private cloud; the robots grasp the boxes and put them in the warehouse separately. Since the edge nodes work independently, the competition resolution algorithm for the MAS that was introduced in Section 4.4 is also appropriate for them. In case one edge node fails to work or is overloaded, the tasks from this node will be handed over to another idle node intelligently. Through this method, the tasks are accomplished with a self-organized mechanism.
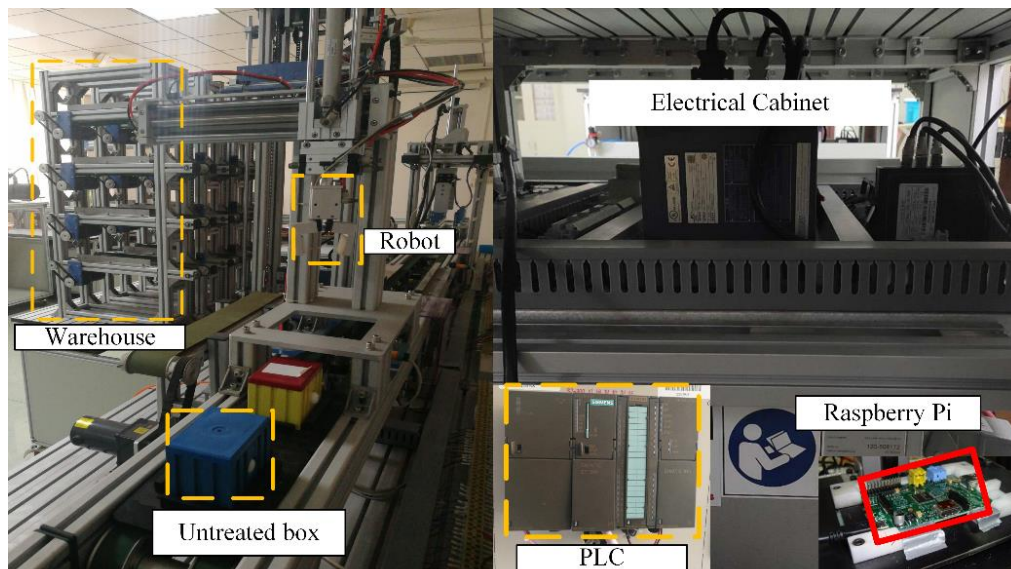
Figure 11. Experimental platform of the case study.

In the case study, the proposed self-organized mechanism is compared with the central handling processor, i.e., the central allocation mechanism. The estimation indicators are the robot operation time and the average upload speed of the backbone network speed. The results are shown in Figure 12. As can be seen from Figure 12a, with the increment of order quantity, the self-organized mechanism has shown more efficient characteristics than the central allocation mechanism. In Figure 12b, after the production line becomes stable, the upload speed of the backbone network based on the central allocation mechanism has reached 6–8 Mb/s. This value has been reduced by more than 50% by the self-organized mechanism. It indicates that the MAS with the competition resolution algorithm can schedule the tasks dynamically in an IoT-based environment. In the case study, Raspberry Pi is granted permission to implement the data fusion and information interaction with the production partners. Owning to the deployment of the edge network and MAS, the service delay is reduced, and the data transmission congestion is avoided.
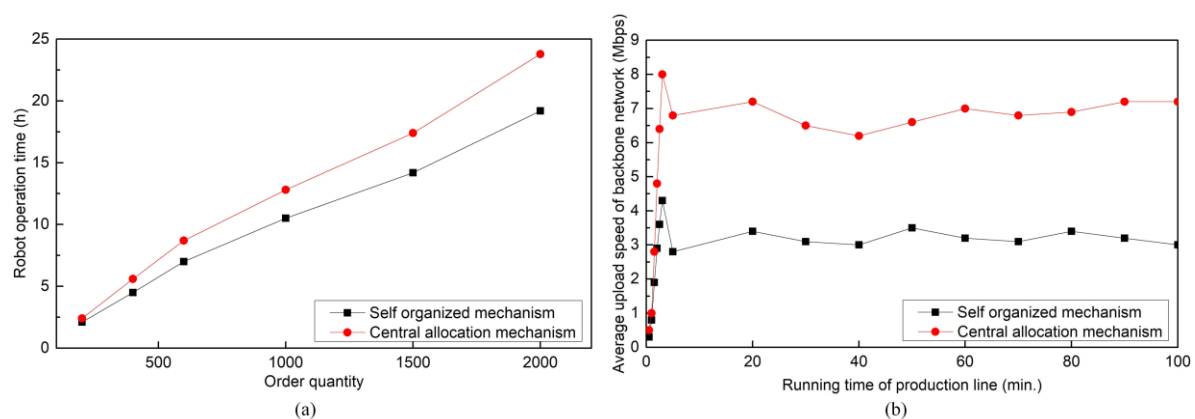
Figure 12. Experimental comparison of the task allocation mechanisms using indicators: (**a**) operation time; (**b**) average upload speed.

5.3. Case Study 3: Conflicts Resolution

This example is designed to verify the feasibility and efficiency of the FCEM introduced in Section 4.4. Suppose that two MASs bid for a production task. To be short, they are denoted as MAS 01 and MAS 02. The bidding process and evaluation procedure are expressed as follows:

(1) Create a set of MAS *B*: *B* = {b_1, b_2} and a set of evaluation arbiters *E*:

$$E = \{e_1, e_2, e_3, e_4, e_5\} \tag{9}$$

(2) Create a set of evaluation indicators *C*:

$$C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\} \tag{10}$$

where:

c_1: overall equipment effectiveness of the bidding MAS;
c_2: expertise of the bidding MAS;
c_3: load of the bidding MAS;
c_4: economic efficiency of the bidding MAS;
c_5: flexibility of the bidding MAS;
c_6: performance of the bidding MAS;
c_7: stability of the bidding MAS.

(3) The weight coefficient *W* for the evaluation indicator set *C*:

$$W = \{0.25, 0.15, 0.1, 0.2, 0.05, 0.1, 0.15\} \tag{11}$$

(4) The single evaluation matrices for bidding MAS 01 and 02 can be generated as follows:

$$R_1 = \begin{bmatrix} 0.45 & 0.25 & 0.10 & 0.05 \\ 0.10 & 0.50 & 0.00 & 0.20 \\ 0.25 & 0.35 & 0.25 & 0.40 \\ 0.50 & 0.00 & 0.5 & 0.45 \\ 0.45 & 0.15 & 0.10 & 0.20 \\ 0.30 & 0.25 & 0.40 & 0.15 \\ 0.30 & 0.20 & 0.30 & 0.35 \end{bmatrix}, \; R_2 = \begin{bmatrix} 0.30 & 0.35 & 0.20 & 0.40 \\ 0.20 & 0.10 & 0.50 & 0.30 \\ 0.45 & 0.00 & 0.10 & 0.20 \\ 0.15 & 0.15 & 0.00 & 0.45 \\ 0.35 & 0.15 & 0.05 & 0.25 \\ 0.40 & 0.25 & 0.10 & 0.50 \\ 0.50 & 0.30 & 0.25 & 0.10 \end{bmatrix} \tag{12}$$

(5) The fuzzy comprehensive estimation matrices for MAS 01 and MAS 02 can be derived through Equation (7): B_1 = [0.35, 0.235, 0.24, 0.25] and B_2 = [0.3125, 0.21, 0.185, 0.3325].

(6) In accordance with maximum membership principle and Equation (5), the MAS 01 is evaluated as the *best* solution, while MAS 02 is the *better* one. MAS 01 is awarded the tender, and will carry on the production task.

6. Conclusions and Future Work

This study presents a proposal of VME characterized by MAS and edge computing in optimizing the rigid system configuration of a manufacturing environment and its computing mode. The MAS-based framework can improve the operation performance of the manufacturing environment, while edge computing demounted the computing load. Moreover, the communication channel between networks is provided using the edge computing technique. The protocols for agent interaction are designed to create a collaborative operation of the MAS. Finally, the simulation with throughput as an evaluation indicator is implemented to verify the superiority of the proposed edge + MAS architecture over the cloud-based architecture and edge-based architecture. The case study is applied with a prototype platform to show

the effectiveness of the proposed architecture. The resolution algorithm designed for the competition of MASs is also proven to be feasible.

Although the MAS has realized the self-organized mechanism in task allocation, the allocation of computing resources is not a concern in this study. It is better to allocate the computing resource in a dynamic way so that the network system will always keep a dynamic balance during the manufacturing procedure despite any emergencies. Future works aim at designing a flexible architecture that allocates the computing resources and bandwidth dynamically with the view of carrying the cost-down project. Besides, the definition mechanism for agent type is relatively rigid, efforts need to be attempted to make the system adapt to the complicated and changeful environment freely. Moreover, as a research hotspot, the combination of fog computing with a MAS in a manufacturing environment is considered.

Author Contributions: Conceptualization, S.T. and Z.L.; methodology, X.Z., X.L. and R.M.; software, X.Z. and X.L.; validation, X.Z., X.L. and Z.L.; formal analysis, X.Z., X.L. and R.M.; investigation, X.L. and S.T.; writing—original draft preparation, X.Z. and X.L.; writing—review and editing, R.M. and Z.L.; supervision, S.T.; project administration, X.L. and Z.L.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Weyrich, M.; Drews, P. Interactive environment for virtual manufacturing: The virtual workbench. *Comput. Ind.* **1999**, *38*, 5–15. [CrossRef]
2. Duin, H.; Oliveira, M.; Saffarpour, A. A simulation model for virtual manufacturing environments for serious games. In Proceedings of the IEEE Technology Management Conference, Sophia-Antipolis, France, 4–6 June 2016; pp. 1–8.
3. Al-Ahmari, A.M.; Abidi, M.H.; Ahmad, A.; Darmoul, S. Development of a virtual manufacturing assembly simulation system. *Adv. Mech. Eng.* **2016**, *8*. [CrossRef]
4. Lu, Y.; Xu, X. A semantic web-based framework for service composition in a cloud manufacturing environment. *J. Manuf. Syst.* **2017**, *42*, 69–81. [CrossRef]
5. Mishra, N.; Singh, A.; Kumari, S.; Govindan, K.; Ali, S.I. Cloud-based multi-agent architecture for effective planning and scheduling of distributed manufacturing. *Int. J. Prod. Res.* **2017**, *54*, 7115–7128. [CrossRef]
6. Zhang, L.; Wang, Z.; Liu, X. Development of a collaborative 3D virtual monitoring system through integration of cloud computing and multiagent technology. *Adv. Mech. Eng.* **2014**, *2014*, 1–10. [CrossRef]
7. Pissinou, N.; Benton, R.; Bhargyavati, B.; Kurkovsky, S. A Roadmap To The Utilization Of Intelligent Information Agents: Are Intelligent Agents The Link Between The Database And Artificial Intelligence Communities. In Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop, Newport Beach, CA, USA, 4 November 1997; pp. 196–205.
8. Liu, Y.Y.; Hung, M.H.; Lin, Y.C.; Chen, C.C.; Gao, W.L.; Cheng, F.T. A Cloud-based Pluggable Manufacturing Service Scheme for Smart Factory. In Proceedings of the 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), Munich, Germany, 20–24 August 2018; pp. 1040–1045.
9. Fassi, I.; Mottura, S.; Sacco, M.; Boër, C.R. An Approach to Virtual Manufacturing Environment. In Proceedings of the Swiss CAD/CAM Conference, Neuchâtel, Switzerland, 15–17 February 1999.
10. Matsas, E.; Vosniakos, G.C.; Batras, D. Effectiveness and acceptability of a virtual environment for assessing human-robot collaboration in manufacturing. *Int. J. Adv. Manuf. Tech.* **2017**, *92*, 3903–3917. [CrossRef]
11. Cai, X.; Cheng, C.; Marwah, A.; Salem, M.H.; Ear, M. Research on behavior simulation of virtual hand in virtual manufacturing environment. *J. Comput. Aided Des. Comput. Graph.* **2015**, *27*, 499–507. (In Chinese)
12. Vosniakos, G.C.; Gogouvitis, X.V.; Vosniakos, G.C.; Gogouvitis, X.V. Construction of a virtual reality environment for robotic manufacturing cells. *Int. J. Comput. Appl. Tech.* **2014**, *51*, 12–19. [CrossRef]

13. Abidi, M.A.; Lyonnet, B.; Toscano, R.; Chevailler, P. Contribution of Virtual Reality for Lines Production's Simulation in a Lean Manufacturing Environment. In Proceedings of the 6th International Conference on Computer Modeling and Simulation, Amsterdam, The Netherlands, 12–13 February 2015; p. 182.

14. Colledani, M.; Pedrielli, G.; Terka, W.; Urgo, M. Integrated Virtual Platform for Manufacturing Systems Design. In Proceedings of the 46th CIRP Conference on Manufacturing Systems 2013, Setubal, Portugal, 29–30 May 2013; pp. 425–430.

15. Hao, Y.; Helo, P.; Shamsuzzoha, A. Cloud-based data exchange and messaging platform implementation for Virtual Factory environment. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 6–9 December 2015; pp. 426–430.

16. García-Magariño, I.; Gutiérrez, C.; Fuentes-Fernández, R. The INGENIAS development kit: A practical application for crisis-management. In Proceedings of the International Work-Conference on Artificial Neural Networks (IWANN), Salamanca, Spain, 10–12 June 2009; pp. 537–544.

17. Cristalli, C.; Foehr, M.; Jäger, T.; Leitao, P.; Paone, N.; Castellini, P.; Turrin, C.; Schjolberg, I. Integration of process and quality control using multi-agent technology. In Proceedings of the IEEE International Symposium on Industrial Electronics, Taipei, Taiwan, 28–31 May 2013; pp. 1–6.

18. Feng, Q.; Bratukhin, A.; Treytl, A.; Sauter, T. A Flexible Multi-Agent System Architecture for Plant Automation. In Proceedings of the IEEE International Conference on Industrial Informatics, Vienna, Austria, 23–27 June 2007; pp. 1047–1052.

19. Suganuma, T.; Oide, T.; Kitagami, S.; Sugawara, K.; Shiratori, N. Multiagent-based flexible edge computing architecture for IoT. *IEEE Netw.* **2018**, *32*, 16–23. [CrossRef]

20. Barenji, R.V.; Barenji, A.V.; Hashemipour, M. A multi-agent rfid-enabled distributed control system for a flexible manufacturing shop. *Int. J. Adv. Manuf. Tech.* **2014**, *71*, 1773–1791. [CrossRef]

21. Marks, P.; Weyrich, M.; Xuan, L.H.; Fay, A. Agent-based adaptation of automated manufacturing machines. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, Limassol, Cyprus, 12–15 September 2018; pp. 1–8.

22. Durica, L.; Micieta, B.; Bubenik, P.; Binasova, V. Manufacturing multi-agent system with bio-inspired techniques: Codesa-prime. *J. Manuf. Syst.* **2015**, *2015*, 829–837. [CrossRef]

23. García-Magariño, I.; Rougemaille, S.; Fuentes-Fernández, R.; Migeon, F.; Gleizes, M.P.; Gómez-Sanz, J. A tool for generating model transformations by-example in multi-agent systems. In Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Salamanca, Spain, 25–27 March 2009; pp. 70–79.

24. Baotong, C.; Jiafu, W.; Antonio, C.; Di, L.; Haider, A.; Qin, Z. Edge computing in IoT-based manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [CrossRef]

25. Lin, Y.; Kemme, B.; Patinomartinez, M.; Jimenezperis, R. Enhancing Edge Computing with Database Replication. In Proceedings of the IEEE International Symposium on Reliable Distributed Systems, Beijing, China, 10–12 October 2007; pp. 45–54.

26. Lay-Ekuakille, A.; Griffo, G.; Fazio, M.; Villari, M. A distributed edge computing architecture to support sensing and detecting leaks in waterworks based on advanced FDM. *IEEE Sens. J.* **2017**, *17*, 7820–7827. [CrossRef]

27. Zhang, J.; Chen, B.; Zhao, Y.; Cheng, X.; Hu, F. Data security and privacy preserving in edge computing paradigm: Survey and open issues. *IEEE Access* **2018**, *99*, 18209–18237. [CrossRef]

28. Lorenzo, B.; Garcia-Rois, J.; Li, X.; Gonzalez-Castano, J.; Fang, Y. A robust dynamic edge network architecture for the internet of things. *IEEE Netw.* **2018**, *32*, 8–15. [CrossRef]

29. Amjad, A.; Rabby, F.; Sadia, S.; Patwary, M.; Benkhelifa, E. Cognitive Edge Computing based resource allocation framework for Internet of Things. In Proceedings of the Second International Conference on Fog and Mobile Edge Computing, Valencia, Spain, 8–11 May 2017; pp. 194–200.

30. Tran, T.X.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. *IEEE Commun. Mag.* **2017**, *55*, 54–61. [CrossRef]

31. Hsu, R.H.; Lee, J.; Quek, T.Q.; Chen, J.C. Reconfigurable security: Edge-computing-based framework for IoT. *IEEE Netw.* **2018**, *32*, 92–99. [CrossRef]

32. Xiao, P.; Hu, Z.; Liu, D.; Yan, G.; Qu, X. Virtual machine power measuring technique with bounded error in cloud environments. *J. Netw. Comput. Appl.* **2013**, *36*, 818–828. [CrossRef]

33. Angrish, A.; Starly, B.; Lee, Y.S.; Cohen, P.H. A flexible data schema and system architecture for the virtualization of manufacturing machines (vmm). *J. Manuf. Syst.* **2017**, *45*, 236–247. [CrossRef]

34. Georgakopoulos, D.; Jayaraman, P.P.; Fazia, M.; Villari, M.; Ranjan, R. Internet of things and edge cloud computing roadmap for manufacturing. *IEEE Cloud Comput.* **2016**, *3*, 66–73. [CrossRef]

35. Nwana, H.S.; Ndumu, D.T. An introduction to agent technology. *Lect. Notes Artif. Intell.* **1997**, *14*, 3–26. [CrossRef]

36. Bruckner, D.; Stănică, M.P.; Blair, R.; Schriegel, S.; Kehrer, S.; Seewald, M.; Sauter, T. An introduction to opc ua tsn for industrial communication systems. *Proc. IEEE* **2019**, *99*, 1–11. [CrossRef]

37. Ahmed, E.; Yaqoob, I.; Gani, A.; Imran, M.; Guizani, M. Social-aware resource allocation and optimization for D2D communication. *IEEE Wirel Commun. Lett.* **2017**, *24*, 122–129. [CrossRef]

38. Zuo, Z.; Zhou, M. Web Ontology Language OWL and its description logic foundation. In Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies, Chengdu, China, 27–29 August 2003; pp. 157–160.

39. Chen, C.T.; Lin, C.T.; Huang, S.F. A fuzzy approach for supplier evaluation and selection in supply chain management. *Int. J. Prod. Econ.* **2006**, *102*, 289–301. [CrossRef]

40. Liu, X.; Peng, G.; Liu, X.; Hou, Y. Development of a collaborative virtual maintenance environment with agent technology. *J. Manuf. Syst.* **2010**, *29*, 173–181. [CrossRef]