Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 56 (2016) 354 – 359

www.elsevier.com/locate/procedia

9th International Conference on Digital Enterprise Technology - DET 2016 – "Intelligent Manufacturing in the Knowledge Economy Era

Using autonomous intelligence to build a smart shop floor

Dunbing Tang[a,*] , Kun Zheng[a], Haitao Zhang[a] Zelei Sang[a], Zequn Zhang[a], Chao Xu[a], Javier A. Espinosa-Oviedo[b,c] Genoveva Vargas-Solar[d], José-Luis Zechinelli-Martini[e]

[a] *College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[b] *French Mexican Laboratory of Informatics and Automatic Control, Ex-hacienda Sta. Catarina Mártir s/n, 72810, Cholula, Puebla, Mexico*
[c] *Barcelona Supercomputing Centre, C/ Jordi Girona 1-3, 08034 Barcelona, Spain*
[d] *French Council of Scientific Research, LIG-LAFMIA, 681 rue de la Passerelle, 38402 Saint Martin d'Hères, France*
[e] *Fundación Universidad de las Américas Puebla, LAFMIA, Exhacienda Sta. Catarina Mártir s/n 72810, Cholula, Puebla, Mexico*

* Corresponding author. Tel.: +86-025-84892051; fax: +86-025-84891501. *E-mail address:* d.tang@nuaa.edu.cn

Abstract

The vision of smart factory is based on the notion of Industry 4.0 that denotes technologies and concepts related to cyber-physical systems and the Internet of Things (IoT). In smart factories cyber-physical systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the IoT, cyber-physical systems communicate and cooperate with each other in real time. This paper presents a smart factory architecture based on communication and computing layers that embed scheduling mechanisms within a mechanical shop floor. Every physical entity in the shop floor is seen as an autonomous intelligent agent that performs tasks guided by dynamic scheduling functions. A test bed has been set up to show how physical entities can be cooperative and autonomous units that can automatize the shop floor operation processes. The results verify the feasibility and efficiency of proposed method.

Keywords: smart shop floor; agent; scheduling; cyber-physical systems; industry 4.0

1. Introduction

The vision of smart factory is based on the notion of Industry 4.0 that denotes technologies and concepts related to cyber-physical systems (CPS) and the Internet of Things (IoT). In smart factories CPSs monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the IoT, cyber-physical systems communicate and cooperate with each other and humans in real time.

In parallel, the advancement of IoT and CPS brings some challenges to manufacturing systems. Research on manufacturing systems is focused on production scheduling [1-3], production control [4, 5], production management [6], and other aspects of manufacturing industry. And manufacturing systems reveal increasing characteristics of discretization, intelligentization and autonomy. With complication of manufacturing systems, it becomes very difficult to realize these characteristics by traditional technologies. Therefore, how to realize these characteristics in manufacturing systems by using IoT and CPS is a crucial issue, which is the focus of this paper.

Currently, the research of IoT in manufacturing systems is mainly aimed at cloud manufacturing systems [7, 8]. This type of manufacturing model transforms traditional product oriented manufacturing to service oriented manufacturing. Research of CPS in manufacturing systems is primarily focused on the communication between physical entities [9]. It pays more attention to artificial intelligence, adaptivity, self-organization, self-regulation and other aspects of autonomic computing functions embedded in physical systems.

This paper presents a smart factory based on communication and computing layers that embed dynamic scheduling mechanisms within a mechanical shop floor. Every mechanical element in the shop floor is seen as an intelligent agent that performs tasks guided by dynamic scheduling functions. We embed computing power and optimization capabilities into each agent so that it can make decisions to agilely respond to frequent occurrence of unexpected disturbances at shop floor.

The remainder of the paper is organized as follows. Section 2 general describes the architecture of a smart shop floor. It defines three layers and uses information interaction between different levels to realize the dynamic scheduling and rescheduling of the smart shop floor. Section 3 presents communication protocols that act as cooperation and interaction rules through the architecture for automatically piloting the smart shop floor. Section 4 describes a pilot test bed for simulating the proposed smart shop floor. Section 5 concludes the paper.

2. Smart shop floor architecture

The idea beyond our smart shop floor is that CPSs monitor physical processes of shop floor, create a virtual copy of the physical world and make decentralized decisions. Over the IoT, smart shop floor architecture (Figure 1) is built upon physical layer, communication layer and logical layer. The functions of different layers will be described in the following subsections.

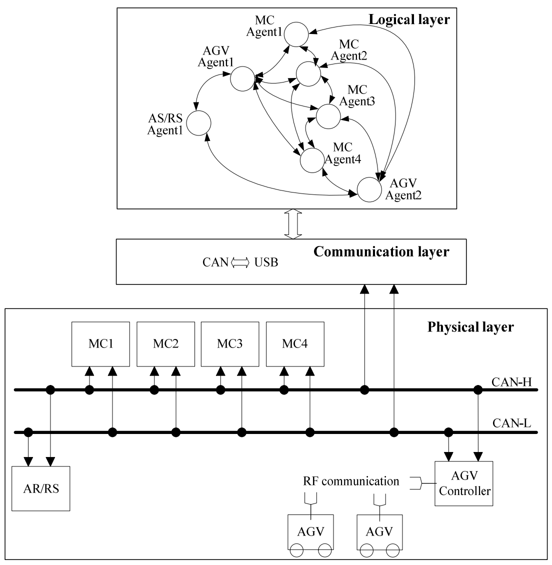

Fig. 1. Smart flow shop architecture.

2.1. Physical layer

Physical layer implements processing and transportation operations in shop floor, and it is composed of manufacturing cells (MCs), automated guided vehicles (AGVs), automated storage/retrieval system (AS/RS), radio frequency (RF) communication and controller area network (CAN) communication. The code developed at this layer is in charge of moving, controlling and monitoring the equipment through RF and CAN communications in the real world. And equipment sends and receives data at this level. The functions of each component are as follows:

- AS/RS handles storage and distribution operations of finished products and raw material.
- MC is composed of machine, robot and buffer. A machine provides several types of processing services; a robot conducts the role of shifting workpieces among machine, buffer, and AGVs; a buffer provides temporary storage service to workpieces.
- AGV deals with transportation operations between MCs and AS/RS.
- AGV controller dispatches AGVs by sending commands, receives feedback messages from AGVs simultaneously.
- RF is a wireless communication mechanism between AGV controller and AGVs.
- CAN is a wired communication mechanism between MCs, AS/RS and AGV controller.

2.2. Logical layer

Logical layer is the logical mapping of physical layer, composed of MC agents, AGV agents and AS/RS agent. In logical layer, every agent is an autonomous and cooperative unit, provided with strong computing power and embedded intelligent scheduling algorithms. Agents receive messages from entities (MCs and AGVs) in physical layer and cooperate with each other to generate the scheduling for the entities. The logical layer provides two types of scheduling services to physical layer: scheduling of operations on machines and scheduling of transportation by AGVs.

For the scheduling of operations on machines, different MC agents produce feasible and optimized plan by competing processing tasks with each other; during the scheduling of transportation tasks, MC agents and AS/RS agent compete AGVs for transporting workpieces, and AGV agents compete transportation tasks. By applying the two scheduling approaches, the logical layer is able to satisfy different scheduling demands of physical layer.

2.3. Communication layer

Communication layer establishes communication between physical layer and logical layer. It connects the two different ports: one port is used to receive and transmit messages in physical layer; another one is used for logical layer. It can transform messages from one layer to the other. It is composed of a set of standards for transforming a data value into zeros and ones that will be transmitted in a network. This layer concerns protocols and standards. In this architecture, a mutual conversion equipment between CAN and USB is used to connect physical layer and logical layer. On this basis, communication layer transforms messages which can be read at different layers. And this transformation implies as follows:

- Receive CAN messages from physical entities and build a message for logical layer that includes complementary information like a time stamp, parameters and other data required by agents.
- Deliver messages to logical layer (in time and order).

- Receive messages from the logical layer and transform them into CAN messages and transmit them to the physical entities.

Physical layer implements actual operations of smart shop floor, and updates status messages to logical layer through communication layer. According status messages, logical layer initiates scheduling or rescheduling between autonomous intelligence agents, and creates feasible scheduling plans. Then logical layer sends scheduling commands to physical entities by using communication layer. In our previous work [10, 11], the physical layer and logical layer have been deeply investigated, and will not be described in this article. In next section, communication protocol in communication layer is detailed.

3. Communication protocol

Communication protocol is the rules and conventions that physical layer and logical layer must observe in communication or service. And it is introduced from two aspects: communication architecture and transformation standards.

3.1. Communication architecture

In order to establish communication between logical layer and physical layer, communication architecture is setup, shown in Figure 2.

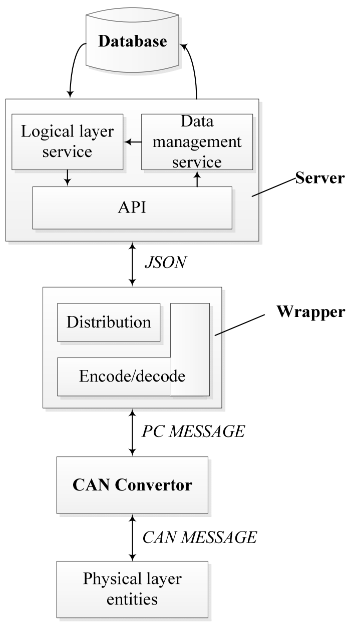

Fig. 2. Communication architecture.

In the architecture, there are four main components: database, server, wrapper and CAN convertor.

- The database component mainly performs the date storage function. The Server can update and read data in the database. In our system, the database is MySQL.
- The server component consists of physical layer service, data management service (DMS) and application interface (API). The physical layer service, coding in java agent development framework (JADE), is provided with several functions, e.g. scheduling, decision-making, etc. And it can also handle the database component. The DMS deals with updated status from physical layer entities. If updated status is a normal state, the DMS updates it into database; if the updated status is a disturbed state, the DMS not only updates it into database but informs the disturbance to the physical layer service. Here, an API is developed to make conversions of message formats between java object and java script object notation (JOSN). All entities in physical layer are mapped to independent Agents, which exist in the physical layer service. The server performs the communication between Agents instead of entities in physical layer.
- The wrapper component converts the message formats between JOSN and PC MESSAGE (the message converted from CAN to USB port, e.g. 00010101). The Wrapper can decode the message format from PC MESSAGE to JOSN directly. In JOSN, some messages with only one destination (e.g. one message from Server to MC1) can be directly encoded to PC MESSAGE format; some messages with multi destinations (e.g. one message from Server to MC1, MC2, and MC3) must be distributed to corresponding sub-message firstly, then they are encoded into PC MESSAGE format.
- The CAN convertor component converts message formats between PC MESSAGE and CAN MESSAGE. In our system, it is a device.

By using this structure, the connection between entities and agents is built. The communication processes are described as follows:

Step(1) Each entity in physical layer sends their status through CAN.

Step(2) The CAN convertor converts message formats from CAN MESSAGE to PC MESSAGE.

Step(3) The Wrapper decodes CANUSB and converts to JOSN.

Step(4) The API of Server converts message formats from JOSN to java object.

Step(5) If messages represent normal state, the DMS updates the state into database and return null (a round communication finished). If messages represent disturbed state (e.g. machine breakdown and machine demands for an AGV), the DMS updates the state into database, and informs the disturbance to the logical layer service at the same time simultaneously.

Step(6) The logical layer service accomplishes rescheduling and/or scheduling processes, and then sends out the results (new plan).

Step(7) The API of server converts the results from java object to JOSN.

Step(8) If the number of message destination is one, the wrapper encodes the message to PC MESSAGE format; if the number of message destination is more than one, the wrapper distributes the message to several sub-messages according to the number of message destination, and then encodes them to PC MESSAGE format.

Step(9) The CAN convertor converts the message(s) to CAN MESSAGE format. (a round communication finished)

Through message format conversion in communication architecture, messages sent by logical layer are transformed to message format that physical layer can read. Moreover, status messages updated by physical layer are also transformed to message format that logical layer can receive.

3.2. Transformation standards

The proposed communication architecture realizes the communication between physical layer and logical layer. Logical entities send messages in CAN MESSAGE format; and agents in logical layer transmit messages in java object format. When messages enter into communication layer from one port, they are transformed into corresponding format messages, and send out from another port. Using this communication standard, agents in logical layer send decision commands to physical entities (MC, AGV and AS/RS) and control them to perform corresponding operations; accordingly physical entities implement operations and collect and update data to logical layer as the decision basis. Taking AGV as an example, the message content sent between AGV entity and AGV agent are described in detail. Message from AGV agent to AGV entity is listed in Table 1; and Message from AGV entity to AGV agent is show in Table 2.

Table 1. Message from AGV agent to AGV entity

Message content	Message example	Description
Type	AGV	Device type
Id	AGV1	Device number
PickUpPoint	MC1	Pick up point of AGV
Destination	MC2	Unload point of AGV
Workpiece	J02	Serial number of workpiece
DelayTime	10	Delay time for executing a task
ConductTime	30	Transportation time of AGV

Table 2. Message from AGV Controller to AGV Agent.

Message content	Message example	Description
Type	AGV	Device type
Id	AGV1	Device number
State	1	State of AGV. "0" is idle state; "1" is busy state; "2" is abnormal state
Workpiece	J01	Serial number of workpiece
xLocation	2	Abscissa of AGV location
yLocation	3	Ordinate of AGV location
Destination	MC1	Unload point of AGV

The message content in Table 1 expressed as Java object is as follows:

```
AgentToEntity A1 = new AgentToEntity ();
A1.setType("AGV");
A1.setId("AGV1");
A1.setPickUpPoint("MC1");
A1.setDestination("MC2");
A1.setWorkpiece ("J02");
A1.setDelayTime(10);
A1.setConductTime(30);
```

AGV agent instantiates message content as "A1" and assigns concrete values to "A1", then sends "A1" to communication layer. After transformation, communication layer sends the transformed CAN message to AGV controller, and CAN message content is shown as follow:

(0x01, 0x01, 0x01, 0x02, 0x02, 0x0A, 0x1E, 0x00)

In above CAN message, fields 1~7 have been occupied and field 8 has not been used. The meanings of corresponding fields are:

"0x01" represents that device type is "AGV";

"0x01" represents that device id of AGV is "1";

"0x01" denotes that pick up point of AGV is "MC1";

"0x02" denotes that unload point of AGV is "MC2";

"0x02" denotes that workpiece transported by AGV is "J02";

"0x0A" indicates that transportation task will be implement by AGV after "10" seconds;

"0x1E" indicates that transportation time of AGV is "30" seconds;

"0x00" represents null field.

When AGV controller receive the above CAN message, it assigns AGV1 to conduct relevant transportation operations by means of RF communication. Accordingly, AGV1 collects and updates its state (table 2) to communication layer. Its form of message transformation is similar to message transformation from AGV agent to AGV entity.

4. Case study

In order to test proposed smart shop floor, a simulation platform is set up, shown in Figure 3. Physical components in test bed have been introduced above, and logical layer and communication layer are operating in the server.

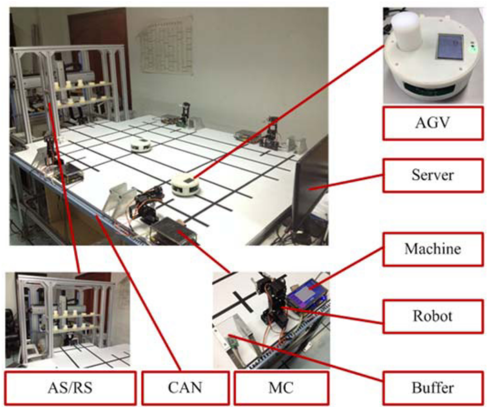

Fig. 3. Simulation platform

According to the numbers of physical entities, 2 AGVs and 4 MCs are set in the experiment. 4 MCs are provided with turning, drilling, grinding and milling function separately. Normal order parameters are listed in Table 3; rush order parameters are presented in Table 4; and transportation time of AGV are listed in Table 5. In the experiment, assumptions and constraints are in the following:

- An AGV can only transport one workpiece at a time;
- When an AGV finishes its transportation task, it stay where it is, pending further orders;
- Each machine can only process one job, and the buffer is large enough for workpieces;
- There are several operations of each job, and each operation needs to be processed on corresponding machine in certain time.

Table 3. Normal order parameters.

Job	Operation	Operation type	Operation time
1	1	turning	12
	2	drilling	24
	3	milling	18
2	1	turning	36
	2	grinding	12
	3	drilling	30
3	1	grinding	18
	2	milling	6
	3	turning	24
4	1	milling	12
	2	drilling	30
5	1	grinding	6
	2	turning	18

Table 4. Rush order parameters.

Job	Operation	Operation type	Operation time
6	1	turning	20
	2	grinding	12
	3	drilling	16
7	1	grinding	18
	2	milling	6
	3	Turning	22
8	1	milling	12
	2	drilling	24

Table 5. Transportation time of AGV.

	AS/RS	MC1	MC2	MC3	MC4
AS/RS	0	8	10	5	8
MC1	6	0	8	5	9
MC2	9	7	0	10	15
MC3	8	5	9	0	7
MC4	10	5	9	13	0

In the experiment, a rush order comes into shop floor at time 70s. Aiming at the disturbance, agents in the logical layer cooperate with each other and generate scheduling decision to physical layer; entities in physical layer implement relevant scheduling commends and update their state to logical layer; communication layer transforms messages between logical layer and physical layer. And the scheduling result is drawn in Gantt chart, shown in Figure 4.

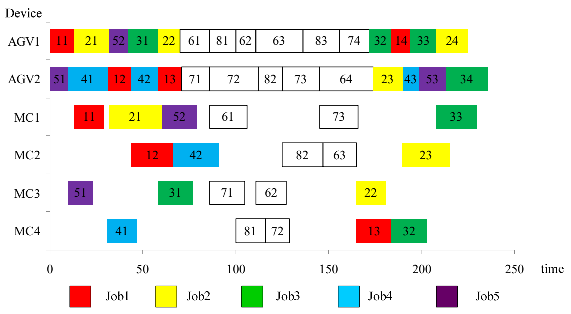

Fig. 4. Gantt chart of simulation result

5. Conclusions

In this paper, a smart shop floor with autonomous intelligence is built. Over the IoT, smart shop floor architecture is built upon physical layer, communication layer and logical layer. On this basis, communication between physical layer and physical layer are established. Using message transformation functions of communication layer, scheduling decision made by autonomous intelligence agents in logical layer can be transmitted to physical entities; physical layer executes scheduling commands and updates status messages to logical layer simultaneously. Therefore, dynamic scheduling of smart shop floor is realized. The feasibility and efficiency of the proposed smart shop floor are verified through the experiment, which is carried out on the simulation platform.

Acknowledgements

This research was sponsored by National Science Foundation of China (NSFC) under Grant No. 51175262 and No. 51575264. This research was also sponsored by the CASES project which is supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme under the grant agreement No. 294931.

References

[1] Burnwal S, Deb S. Scheduling optimization of flexible manufacturing system using cuckoo search-based approach. The International Journal of Advanced Manufacturing Technology 2013; 64(5-8): 951-959.

[2] Gen M, Lin L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing 2014; 25(5): 849-866.

[3] Erol R, Sahin C, Baykasoglu A, et al. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. Applied Soft Computing 2012; 12(6): 1720-1732.

[4] Ounnar F, Pujo P. Pull control for Job Shop: Holonic Manufacturing System approach using multicriteria decision-making. Journal of Intelligent Manufacturing 2012; 23(1): 141-153.

[5] Njike A N, Pellerin R, Kenne J P. Simultaneous control of maintenance and production rates of a manufacturing system with defective products. Journal of Intelligent Manufacturing 2012; 23(2): 323-332.

[6] Ngai E W T, Chau D C K, Poon J K L, et al. Implementing an RFID-based manufacturing process management system: Lessons learned and success factors. Journal of Engineering and Technology Management 2012; 29(1): 112-130.

[7] Li B H, Zhang L, Wang S L, et al. Cloud manufacturing: a new service-oriented networked manufacturing model. Computer Integrated Manufacturing Systems 2010; 16(1): 1-7.

[8] Tao F, Cheng Y, Da Xu L, et al. CCIoT-CMfg: cloud computing and Internet of things-based cloud manufacturing service system. Industrial Informatics, IEEE Transactions on 2014; 10(2): 1435-1442.

[9] Shi J, Wan J, Yan H, et al. A survey of cyber-physical systems. Wireless Communications and Signal Processing (WCSP), 2011 International Conference on. IEEE, 2011: 1-6.

[10] Zheng K, Tang D, Giret A, et al. Dynamic shop floor re-scheduling approach inspired by a neuroendocrine regulation mechanism[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 2015; 229(S1): 121-134.

[11] Tang D, Gu W, Wang L, et al. A neuroendocrine-inspired approach for adaptive manufacturing system control. International Journal of Production Research 2011; 49(5): 1255-1268.