International Journal of Modeling and Optimization, Vol. 14, No. 1, 2024

Designing a Digital-twin Based Dashboard System Framework for a Flexible GPU Card Assembly Line

Tsung-Lun Li[1,*], Kung-Jeng Wang[1], Agustina Eunike[1], Phuc Hong Nguyen[2], and Jing-Ming Chiu[3]

[1]Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan
[2]Department of Industrial Management, Can Tho University, Can Tho City, Viet Nam
[3]Institute for Information Industry, Government, Taipei, Taiwan
Email: welcome.lenlee@gmail.com (T-L. L); Kjwang@mail.ntust.edu.tw (K-J. W.); agustina.eunike@ub.ac.id (A. E.); nguyenhongphuc@ctu.edu.vn (P. H. N.); jmchiu@iii.org.tw (J-M. C.)
*Corresponding author

Manuscript received May 30, 2023; revised June 25, 2023; accepted July 23, 2023; published March 7, 2024

***Abstract*—The current dashboard system is limited in communication capability. This study proposed a digital twin dashboard system framework. To realize the digital-twin-based dashboard system, for the field application of this research laboratory, a simple contract net-based communication protocol is designed to realize a dashboard system with communication capabilities. The digital twin-based dashboard system is successfully implemented in a flexible GPU card assembly line. Enabling each digital twin module system to interact and operate flexibly to demonstrate the relationship between the digital twins and the physical modules.**

***Keywords*—dashboard system, digital twin, flexible assembly line, human-machine interface**

I. INTRODUCTION

The main function of the dashboard system in the current industry is to display data and monitor the environment, and it has several basic technologies, such as management, data analysis, Human-Machine Interface (HMI), network and communication, computing security and security, etc. technologies [1]. However, the current dashboard system is limited to collecting data on a platform for analysis, and cannot further connect the machines and equipment to communicate, and then display the results of the communication on the dashboard, so it still lacks coordination, flexibility and scalability.

Smart physical production requires digitalization technology to help make real-time decisions, for example, to visualize key data [2]. In smart production systems, digital twin technology can use sensors to build physical devices in virtual spaces to reflect the life cycle of physical machines and products [3]. Through digital twin technology, the behavior, rules, status and logic of physical objects (such as personnel, products, equipment, production lines, and environment) can be modeled in virtual space, and the models can communicate with each other [4], and further optimization solutions can be proposed [5].

The present study is motivated by the background virtual space in the current dashboard system does not have communication capability. The realization of a dashboard system with communication capabilities can increase the coordination, flexibility and scalability of intelligent physical production. This study aims to propose a digital twin dashboard system framework and to develop a prototype for feasibility analysis. Through literature and industry surveys, the potential gaps in the application of digital twin technology to the dashboard system were determined, and a design prototype and framework were proposed accordingly.

II. LITERATURE REVIEW

A. Modern Dashboard Systems

Dashboard system is a visual performance management tool [6] that visualizes information about important organizational goals, such as requirements, direction, and conducts action guidance, as well as support and improves on-line decision making [7].

The dashboard provides different management functions. For example, it can provide a storage device to monitor [8], manage energy operation [9], control devices [10], manage orders and machines [11], enhance human insight and long-term/short-term decision making [12, 13], and control scenes and databases [14].

B. Intelligent Agent and Digital Twin Technology

In the context of digitalization, smart products are seen as a central entity in the production system that selects the best production resources and organizes further tasks through specific process steps [15]. A multi-agent system consists of multiple agents with six core attributes [16]: distribution, adaptability, flexibility, scalability, leanness, and resiliency. Each entity is represented by an intelligent agent and can be run dynamically at runtime [17].

"Intelligent agent" is usually used to indicate the names of programs with different characteristics and functions, and has four characteristics, namely autonomy, social ability, initiative and responsiveness [18]. The intelligent agent technology has naturally evolved into a multi-agent system (MAS) over time [19]. A multi-agent system is composed of multiple intelligent agents connected to each other, and the behavior of each agent is also affected by the behavior of neighboring agents [20].

The agent system technology is entity dynamic, which can divide complex tasks, assign tasks to the most suitable agents, and automatically arrange the agents [21]. According to customer requirements of product orders, a multi-agent system for manufacturing control can select the machine or resources to process it and achieve intelligent production [15].

Digital twin is currently one of the most promising digital technologies, which can support digital transformation and decision-making in many industries [22]. Digital twin technology is an emerging technology derived from intelligent agent technology and is a core competency in

DOI: 10.7763/IJMO.2024.V14.846

Industry 4.0 [23], and integrates the physical and digital world [24]. It provides comprehensive physical and functional descriptions of compartments, products, or systems, and can be expanded independently and updated automatically [3].

Digital twins represent physical devices or components by integrating data and simulation, and this technology has both static and dynamic information [25]. The static information includes geometric sizes, lists of materials, and procedures, whereas the dynamic information includes information on the product life cycle that changes over time [26]. Through a large amount of information from various devices, it can be continuously updated and visualized to identify the equipment status of current and future [27]. At present, the implementation of digital twins mostly revolves around data and communication interfaces, emphasizing that the communication between Digital twin models requires more effective and safer communication protocols. [28].

However, a digital twin is not a complete model of a physical product; rather, it is a series of digital data and simulated models with different purposes [27]. Digital twins can be used both to predict the result of certain purposes and to predict product and service behaviors [29]. Digital twin has proved to be a way to integrate the physical world and the virtual world of manufacturing [30], and have three levels: monitor, predict, and decide [31]. It can build virtual models of physical equipment, production lines, factories, and people, as well as related data repositories [32–35] and record its operation process and decision-making information, further use of digital twin system monitoring, control and simulation of physical real status [36].

C. *Research Gaps in Dashboard Systems*

According to the previous research [1], the technology related to digital twins has various effects and reached saturation. However, there are still several applications of technological impact that are still in the early stages, and there are even gaps in digital twin technology, and these gaps may constitute many potential impacts. For example, HMI technology solved communication capability and disruption, diagnostics, and prevention; optimization and machine learning system solved problems in pattern recognition and classification; wearable device technology solved pattern recognition and classification.

III. A DIGITAL TWIN BASED DASHBOARD SYSTEM PROTOCOL DESIGNING

A. *The Target System – A Flexible GPU Card Assembly Line*

Each physical device has a corresponding virtual agent. Table 1 describes the number of resources and assigned tasks for each agent. Operators and robotic arms are defined as Active agents, which dominate the sorting rules. The screw machine, conveyor belt, Automatic Optical Inspection Machine (AOI), GPU cards, and storage area are defined as passive agents, and the rule is first come first served. In the Flexible Assembly Line (FAL), the active agent is responsible for sequencing the parts for the next action.

Table 1. Definition of flexible assembly line unit (agent)

Agents	**Label**	**Role**	**Task**	**Number**
Operator	H1	Active agent	Pick and place Screwing Rework	1
Robotic arm	Nexcom (R2) TM (R3)	Active agent	Pick and place	2
Screw machine	R1	Passive Agent	Screwing	1
Conveyor	Con	Passive Agent	Transport objects	1
Automatic Optical Inspection Machine (AOI)	AOI	Passive Agent	Optical module for quality check	1
GPU card	P1	Passive Agent	Production object	*n*
Storage cache	Storage	Passive Agent	Finished good end point	*n*

The digital twin dashboard system backend model consists of several agents, including the screw machine (R1), industrial robotic arm (R2), collaborative robotic arm (R3), Operator (H1), conveyor belt (Con), and AOI. The Relevant Scenario of dynamic production line process is shown in Fig. 1. First, the GPU motherboard and backplane materials are supplemented by H1 to the R1 machine, and R1 performs the locking operation of the two materials. At the same time, the R2 machine receives the specified GPU model, and clamps the GPU's fan to a relative position. H1 takes the assembled mainboard and bottom plate from R1, and locks it directly on the fan. After the locking is completed, R2 places the finished GPU on the conveyor belt and sends it to R3 for AOI inspection. If there is a special situation of shipment (such as priority shipment, etc.) before the AOI inspection, H1 will carry out the program planning according to the situation before the inspection, and then send it to R3 to complete the AOI inspection. Finally, R3 grips the GPU finished product to a relative position for packaging or human inspection according to the inspection results.

B. *Definition of Digital Twin Components*

The digital twin components are divided into four, namely the operator, screw machine, industrial robot and automatic optical inspection. The description of each component is shown in Table 2.

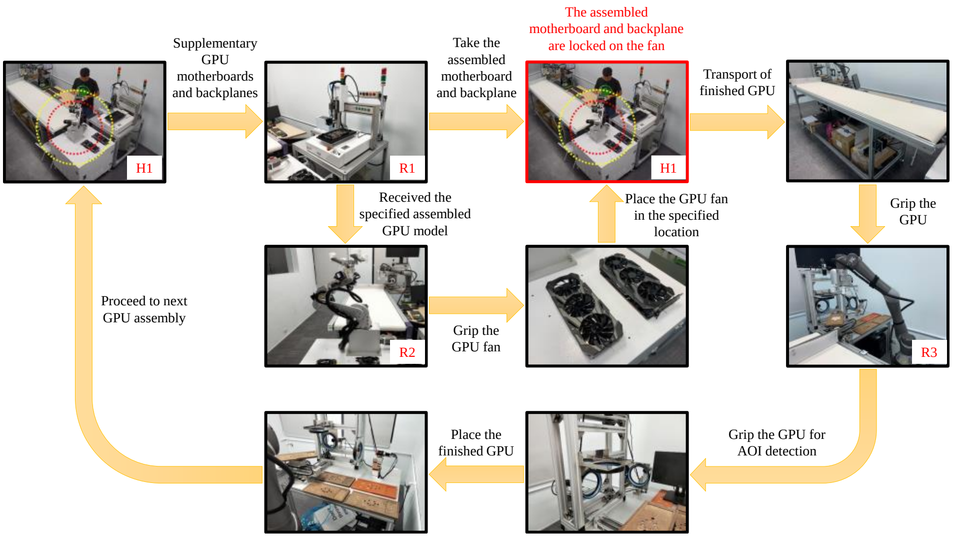


Fig. 1. Scenario of the flexible GPU card assembly line.

Table 2. The description of each component

Component	Label	Description
Operator	H1	It is a human operator. Responsible for picking up and replenishing materials, and locking the GPU card motherboard and fan.
Screw machine	R1	It is an automatic screw-locking robot. Responsible for locking the motherboard and protective bottom plate of the GPU card. There are different numbers of screws and hole position settings for different types of GPU cards.
Industrial robot	R2	It is a six-axis industrial robot. Responsible for clamping the GPU card fan in a fixed position, moving it and fixing it in the corresponding position for H1 to assemble. When moving, follow the man-machine safety distance setting.
Automatic optical inspection	AOI	It is an automatic optical inspection module. Responsible for GPU card defect detection, using machine vision technology to capture images in a non-contact manner, and analyze them with algorithms to determine whether the detection target has defects.

C. Proposed Communication Protocol Among Digital Twins

This research designed a communication protocol that can be operated in the background of the dashboard. It contains multiple digital twin models, and each digital twin can communicate with each other through the rules of the communication protocol. By integrating the dashboard with the communication function, the digital twins in the back-end virtual space of the dashboard system can communicate with each other, to realize the HMI technology to solve the gaps in communication capabilities and interruption, diagnosis, and prevention. First, this study defines the digital twin component in the communication protocol. Then, the communication protocol of each digital twin component is introduced. Finally, a dashboard system framework is proposed based on the communication protocol.

The complexity of the communication protocol has many characteristics, including flow control and communication direction. The simplest protocol is one with no flow or error control. It is also a one-way protocol in which data is transmitted in only one direction (from sender to receiver). In the MAS protocol, the communication process and the calculation information process can be promoted. However, local capabilities and unpredictable interruptions should be considered in a robust protocol, while maintaining a low-cost and efficient process.

Contract Net Protocol (CNP) is a one-way auction model, usually used in the research and practice of multi-agent control systems. CNP controls the cooperation of two roles related to task execution through an interactive auction process [37]. The agent can be a manager responsible for monitoring and proving the results of the task, or a contractor responsible for performing the task.

This study designs a robot communication protocol. This robot communication protocol is a set of rules for formatting and processing data, which controls how each node in the multi-agent system achieves the desired behavior. Each node must communicate with other nodes so that they can reach a consensus in accordance with the rules of resource allocation and access to resources. The simplest protocol is only one-way communication from the sender to the receiver, and the protocol is only decided by one party.

The contract network protocol means to guide the interaction between the auction customers (C nodes) by assigning tasks to the scheduler (Human + R1, R2, and AOI). In this round of auctions, communication occurs between the client and the dispatcher and there is no communication between the dispatchers. The CNP communication related to handling scheduling problems in smart assembly lines is shown in Fig. 2.

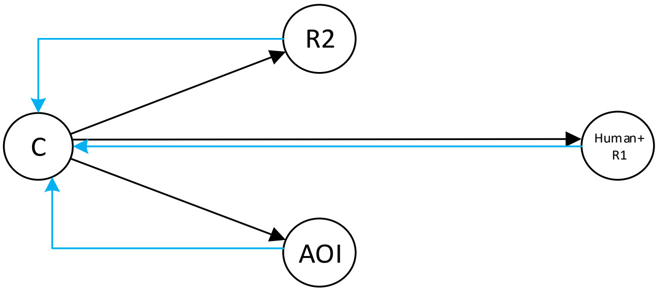


Fig. 2. Contract network protocol communication diagram.

The CNP process is divided into four steps (as shown in Fig. 3), including job announcement, local planning and submission, plan evaluation and plan confirmation. Based on the implementation of the CNP scheduling process, this plan has designed three types of events. The events are (i) arrival of new jobs (Table 3a), (ii) change due date (Table 3b) and (iii) change processing time (Table 3c) to adapt to the production line environment. In addition, Table 3 introduces the description of the steps, including the transfer of message content between agents and the calculation process performed by each agent.

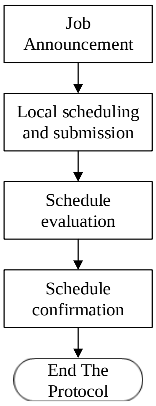


Fig. 3. CNP program.

Table 3. Event definition
A. New work event

Step	Description
Job announcement	The customer issues a new task and broadcasts it to the scheduler: Operator (H1), Nexcom Robot (R2) and AOI. The announcement information mainly includes the job number, GPU type and due date.
Local planning and submission	Each scheduler evaluates its ability to process jobs, and uses heuristic scheduling rules (FCFS, SPT, and EDD) based on personal information to obtain the best local schedule (that is, estimated job completion time). Then, the scheduler submits the schedule to the customer.
Plan evaluation	The customer evaluates the scheduler by calculating the estimated total completion time and making a decision.
Plan confirmation	Confirm the contract by granting the dispatcher the assignment.

B. Change due date event

Step	Description
Job announcement	The customer publishes the due date of the job revision in the queue and broadcasts the due date to the scheduler: Operator (H1), Nexcom Robot (R2) and AOI. The announcement information mainly includes the job number, GPU type, and deadline for revision.
Local planning and submission	Each scheduler will evaluate its ability to process jobs based on the revised deadline, and perform a rescheduling process based on personal information to obtain the revised best local schedule (that is, the new estimated job completion time). Subsequently, the scheduler submits information including the job number and estimated completion time of each to the customer.
Plan evaluation	The client evaluates the schedule by calculating the total revision estimated time to completion and making a decision.
Plan confirmation	Confirm the contract by granting the dispatcher the assignment.

C. Change processing time event

Step	Description
Job announcement	-
Local planning and submission	When the scheduler (operator (H1), Nexcom Robot (R2), and AOI) cannot complete the confirmed work, it will modify the local schedule (that is, the estimated work completion time) based on its personal information. Subsequently, the scheduler began to submit the revision information of the job number and the estimated completion time of the revision to the client.
Plan evaluation	The client evaluates the revised schedule by calculating the estimated total completion time and making a decision.
Plan confirmation	Confirm the contract by granting the dispatcher to modify the assignment.

The client broadcasts job announcements to all schedulers triggered by new jobs or changed due dates. The scheduler receives messages from the client and checks its capabilities based on local information (including jobs in the queue, processing time, and waiting time) to process the local scheduling algorithm. The local scheduling algorithm will get a series of jobs, as well as the completion time of each job, and propose to the customer. In addition, all available schedulers will bid to the customer, and the customer will evaluate the bid to select the overall best total completion time. The customer then sends the contract and confirmed work to the dispatcher. The dispatcher accepts the contract and performs the work. However, when the scheduler interrupts the contract due to a change in processing time, it will trigger a new event, and the scheduler will start sending a message to the customer, and then the customer will conduct another auction for the work on the contract. In Contract Net Protocol, the interaction between the customer agent acting as the manager and the dispatcher agent acting as the contractor is shown in Fig. 4.

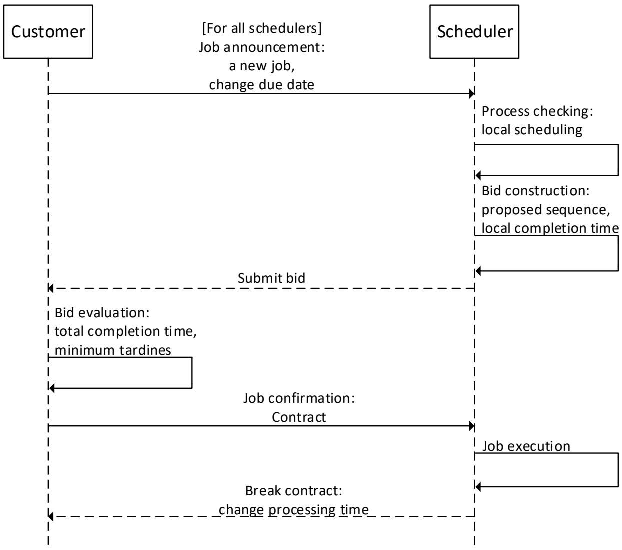


Fig. 4. The interaction diagram between the client and the dispatcher in the Contract Net protocol.

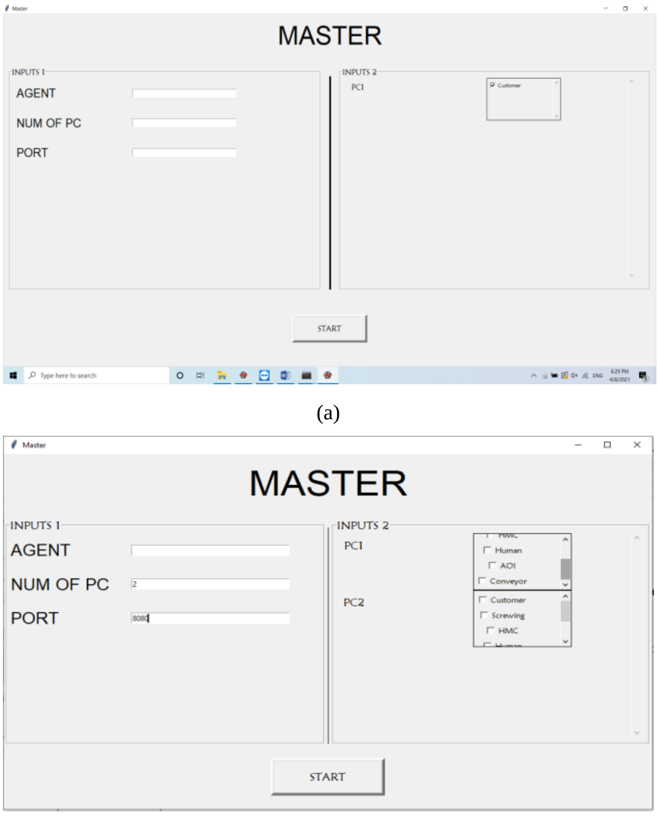
D. Digital Twin Based Dashboard System Implementation

Based on the development of the Contract Net protocol, the agent interaction provides an effective and realistic negotiation mechanism. Especially in multi-agent-based production and manufacturing systems, the contract network protocol can flexibly and elastically respond to different situations, and realize dynamic resource allocation. Through the Contract Net protocol, this research proposes a dashboard system based on digital twins, and in this context presents the system interaction and flexible operation of each digital twin module.

In terms of python development, this study designs a GUI interface of a set of instant programs, which is attached to each agent function. The GUI is divided into three parts, namely the master GUI, the customer GUI, and the scheduler

GUI. First of all, the Master GUI is the interface connected to the main program. Its main function is to divide the PCs we want to use and assign their own agents to these PCs, as shown in Figure 5(a).

When the program is running (Fig. 5(b)), the interface is divided into input 1 and input 2. In input 1, there are 3 input boxes, which are the agent, the number of PCs and the port number. The agent input box can place each agent or scheduler associated with the process. During the input process, you need to enter an agent required for operation, which will save the agent name we entered. Number of PCs Enter the number of PCs that will be used. Port is to enter the port number associated with the PC. On Input 2, the location of each agent that has been entered in Input 1 is displayed. Master GUI operation steps are as follows: (1) Enter the name of the agent during this process and press Enter. (2) Enter the processing time for each type of agent. (3) Enter the number of PCs to be used, and then press Enter. When the first three steps are completed. (4) When the first three steps are completed, enter the connected initial port number. (5) Allocate and designate an agent in each PC. (6) Click the Start button.



(a)

NUM OF PC 2

PORT 8080

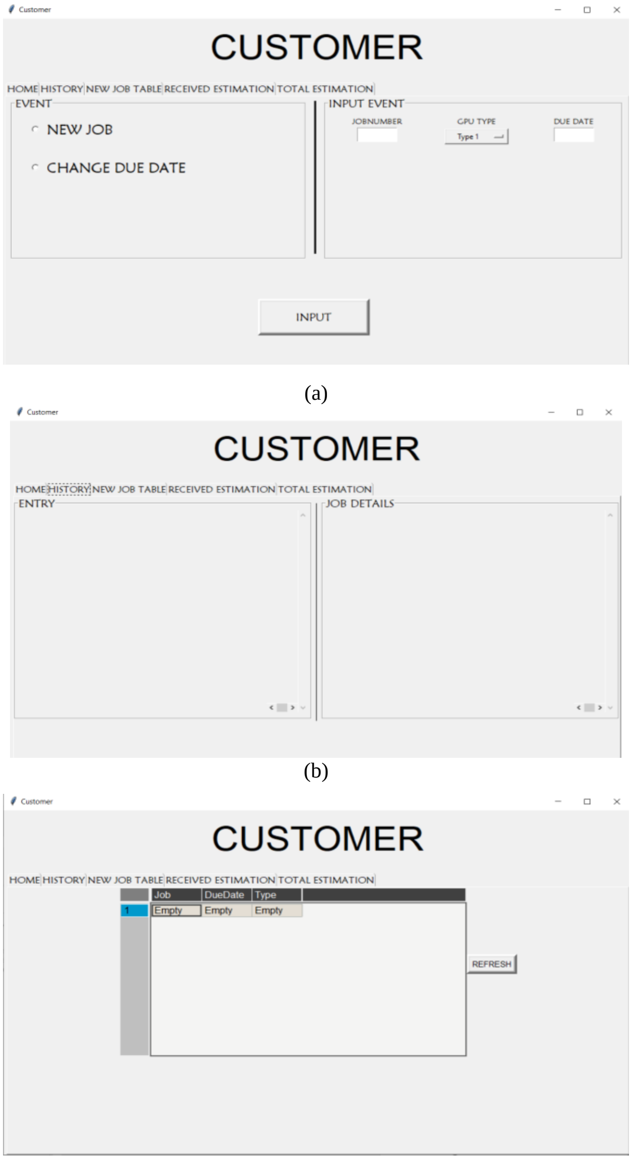
(b)

Fig. 5. Master GUI, (a) Initial Master GUI, (b) Master GUI after agent input.

In terms of Customer GUI, the customer GUI is the connection interface of the client agent, which has five function options. The main function is to enter new job events and change due date events, and then display the output of the system. The main interface of the Customer GUI is shown in Appendix Fig. 6(a). Introduction of Customer GUI function options: (1) New job: If the new event is a new task, click the button. (2) Change due date: If the new event is due date change, click the button. (3) Job number: Enter the job number of the event. (4) GPU Type: Select the type of GPU. (5) Due Date: Enter the due date for the relevant work. (6) Home page option: Open the home page. (7) History record option: Open the history record. (8) New Job Table option: Open a new job Excel file. (9) Received estimate option: Open the received estimate Excel file. (10) Total estimate option: Open the total estimate Excel file.

The History option will display everything related to the new event, while the job details will show the output for that job, such as job completion time, delay time, and job number, as shown in Fig. 6(b).

The new job table option, received estimate option and total estimate option will have the same interface, as shown in Fig. 6(c). If any changes or updates occur, the Excel file can be updated via the refresh button. Customer GUI operation steps are as follows: (1) Select a new event (new job or change due date). (2) Enter the job number. If it is a new job, the job number will be ignored. (3) Select the GPU type. (4) Enter the new due date for the assignment. (5) Click the Enter button.



(a)

(b)

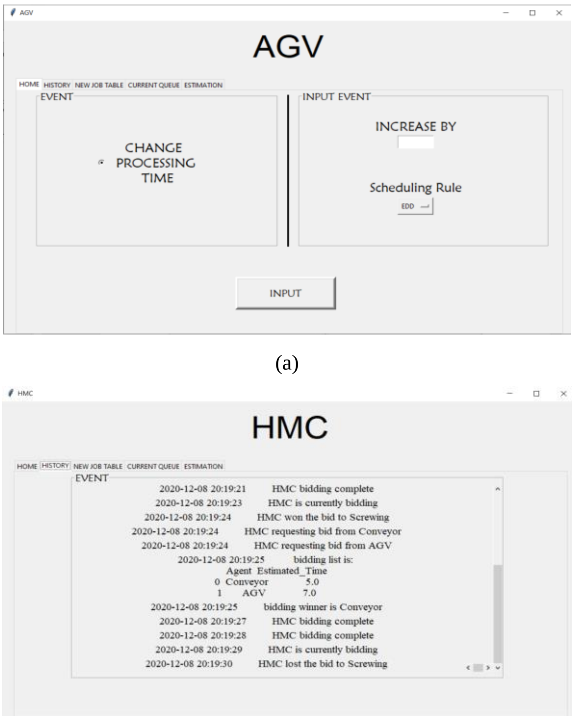
	Job	DueDate	Type
1	Empty	Empty	Empty

(c)

Fig. 6. Customer GUI, (a) Customer GUI, (b) Customer GUI history option, (c) Customer GUI table options

In terms of the Scheduler GUI, it is connectable to every scheduler function and has the same five options as the Customer GUI. The only difference is the home page option. Because in the Scheduler GUI, the new event is to change the processing time of that scheduler agent. The home page option for the Scheduler GUI is shown in Figure 7(a). Introduction of Scheduler GUI function options: (1) Change processing time: Select the event to change the processing time. (2) Increase: Change in input processing time. (3) Scheduling rule: select the new scheduling rule of the agent.

The history option of the scheduler GUI can display all items. If the protocol is advanced, it will also display the bid history and results for each scheduling agent, as shown in Figure 7(b). Scheduler GUI operation steps are as follows: (1) Select the event to change the processing time. (2) Enter the value to be added to the current processing time. For example: if the speed will be 5 slower, enter 5; if the speed will be 5 faster, enter -5. (3) Select a new scheduling rule (EDD, SPT, FIFO or smart scheduling). (4) Click the input button.



(a)

```
2020-12-08 20:19:21        HMC bidding complete
2020-12-08 20:19:23        HMC is currently bidding
2020-12-08 20:19:24        HMC won the bid to Screwing
2020-12-08 20:19:24        HMC requesting bid from Conveyor
2020-12-08 20:19:24        HMC requesting bid from AGV
2020-12-08 20:19:25        bidding list is:
      Agent  Estimated_Time
    0  Conveyor       5.0
    1      AGV        7.0
2020-12-08 20:19:25        bidding winner is Conveyor
2020-12-08 20:19:27        HMC bidding complete
2020-12-08 20:19:28        HMC bidding complete
2020-12-08 20:19:29        HMC is currently bidding
2020-12-08 20:19:30        HMC lost the bid to Screwing
```

(b)

Fig. 7. Scheduler GUI; (a) Scheduler GUI home page, (b) Scheduler GUI history option.

IV. CONCLUSION

This study proposed a digital twin dashboard system framework and developed a prototype for feasibility analysis. The digital twin components are divided into four, namely the operator, screw machine, industrial robot, and automatic optical inspection in a flexible GPU card assembly line. This study designed a communication protocol that can run in the background of the dashboard, which integrates multiple digital twin models, whereas each digital twin can communicate with each other through the rules of the communication protocol.

For the on-site dashboard application of the laboratory, and combined with the digital twin dynamic prototype simulation system, the environment of the digital twin dashboard system is formed. The GUI interface of the real-time program is divided into three parts (i) Master GUI (ii) Client GUI (iii) Scheduler GUI, which can be controlled experimental environment for verification. Experiments have found that the digital twin module can communicate more efficiently, reduce personnel operation time, and help managers solve complex scheduling problems, thereby realizing a dashboard system with communication capabilities.

The contribution of this study lies in two aspects. First, this study proposes the communication technology of the dashboard system based on the digital twin. Through the development of the Contract Net protocol, agent interaction provides an effective and realistic negotiation mechanism, enabling each digital twin module system to interact and operate flexibly. Second, this study presents a landscape of digital twin dashboards. Through the landscape of the digital twin dashboard, we can clearly understand the relationship and operation between the digital twin module and the physical environment, and provide a basis for the future digital twin technology design.

The scope of this study is limited to dashboard communication techniques. At present, the development of the digital twin dashboard system can only present the simple communication technology of each module in the background of the dashboard, without considering other technologies related to digital twins. On this basis, future research can consider more digital twin technologies, such as wearable devices and other technologies. Then a scenario of an advanced communication protocol is designed for detailed elaboration and development, which further improves the framework of the digital twin dashboard system in this study.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Tsung-Lun Li: Writing- Original draft preparation, Experiments, Data analysis. Kung-Jeng Wang: Conceptualization, Methodology, Supervision. Phuc Hong Nguyen: Coding. Agustina Eunike: Coding. Jing-Ming Chiu: Conceptualization, Supervision.

ACKNOWLEDGMENT

Thanks to Kung-Jeng Wang, Phuc Hong Nguyen, Agustina Eunike and Jing-Ming Chiu for assisting in this research. Thanks to the International Conference on Mechanical Manufacturing and Industrial Engineering for giving me the opportunity to present our research.

REFERENCES

[1] K. J. Wang, T. L. Lee, and Y. Hsu, "Revolution on digital twin technology – A patent research approach," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 11, pp. 4687–4704, 2020.

[2] G. Sedrakyan, E. Mannens, and K. Verbert, "Guiding the choice of learning dashboard visualizations: Linking dashboard design and data visualization concepts," *Journal of Computer Languages*, vol. 50, pp. 19–38, 2019.

[3] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Annals*, vol. 66, no. 1, pp. 141–144, 2017.

[4] Z. Chen and L. Huang, "Digital twins for information-sharing in remanufacturing supply chain: A review," *Energy*, 119712, 2020.

[5] P. Wang and M. Luo, "A digital twin-based big data virtual and real fusion learning reference framework supported by industrial internet towards smart manufacturing," *Journal of Manufacturing Systems*, vol. 58, pp. 16–32, 2021.

[6] C. Bachechi, L. Po, and F. Rollo, "Big data analytics and visualization in traffic monitoring," *Big Data Research*, vol. 27, 2020.

[7] S. Vilarinho, I. Lopes, and S. Sousa, "Developing dashboards for SMEs to improve performance of productive equipment and processes," *Journal of Industrial Information Integration*, vol. 12, pp. 13–22, 2018.

[8] S. M. Shaker, "Lessons learned from war room designs and implementations," Evidence Based Research Inc Vienna VA, 2020.

[9] A. A. Kim, Y. Sunitiyoso, and L. A. Medal, "Understanding facility management decision making for energy efficiency efforts for buildings at a higher education institution," *Energy and Buildings*, vol. 199, pp. 197–215, 2019.

[10] F. Shrouf and G. Miragliotta, "Energy management based on Internet of Things: practices and framework for adoption in production management," *Journal of Cleaner Production*, vol. 100, pp. 235–246, 2015.

[11] D. Ivanov, "Disruption tails and revival policies: A simulation analysis of supply chain design and production-ordering systems in the recovery and post-disruption periods," *Computers and Industrial Engineering*, vol. 127, pp. 558–570, 2019.

[12] T. Boukherroub, S. D'amours, and M. Rönnqvist, "Sustainable forest management using decision theaters: Rethinking participatory planning," *Journal of Cleaner Production*, vol. 179, pp. 567–580, 2018.

[13] C. Kumsap, V. Mungkung, I. Amatacheewa, and T. Thanasomboon, "Conceptualization of military's common operation picture for the enhancement of disaster preparedness and response during Emergency and communication blackout," *Procedia Engineering*, vol. 212, pp. 1241–1248, 2018.

[14] M. Sunners, D. Dawson, and P. Sydenham, "Building an interactive blackboard framework," *Knowledge-Based Systems*, vol. 7, no. 3, pp. 199–206, 1994.

[15] M. Klein *et al*., "A negotiation-based approach for agent-based production scheduling," *Procedia Manufacturing*, vol. 17, pp. 334–341, 2018.

[16] A. S. Palau, M. H. Dhada, K. Bakliwal, and A. K. Parlikad, "An industrial multi agent system for real-time distributed collaborative prognostics," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 590–606, 2019.

[17] T. Jung, P. Shah, and M. Weyrich, "Dynamic co-simulation of internet-of-things-components using a multi-agent-system," *Procedia CIRP*, vol. 72, pp. 874–879, 2018.

[18] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[19] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2020.

[20] A. Amato, A. Quarto, and V. D. Lecce, "An application of cyber-physical system and multi-agent technology to demand-side management systems," *Pattern Recognition Letters*, vol. 141, pp. 23–31, 2021.

[21] K. Bakliwal, M. H. Dhada, A. S. Palau, A. K. Parlikad, and B. K. Lad, "A multi agent system architecture to implement collaborative learning for social industrial assets," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 1237–1242, 2018.

[22] E. V. D. Horn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decision Support Systems*, 113524, 2021.

[23] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017.

[24] S. S. Kamble, A. Gunasekaran, H. Parekh, V. Mani, A. Belhadi, and R. Sharma, "Digital twin for sustainable manufacturing supply chains: Current trends, future perspectives, and an implementation framework," *Technological Forecasting and Social Change*, vol. 176, 121448, 2022.

[25] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with automationML and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.

[26] S. Haag and R. Anderl, "Digital twin–Proof of concept," *Manufacturing Letters*, vol. 15, pp. 64–66, 2018.

[27] S. Mathupriya, S. S. Banu, S. Sridhar, and B. Arthi, "Digital twin technology on IoT, industries and other smart environments: A survey," *Materials Today: Proceedings*, 2020.

[28] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *Journal of Manufacturing Systems*, 2020.

[29] S. Boschert and R. Rosen, "Digital twin — The simulation aspect," *Mechatronic Futures*, pp. 59–74, 2016.

[30] Q. Min, Y. Lu, Z. Liu, C. Su, and B. Wang, "Machine learning based digital twin framework for production optimization in petrochemical industry," *International Journal of Information Management*, 2019.

[31] Í. A. Fonseca, H. M. Gaspar, and P. C. Mello, and H. A. U. Sasaki, "A standards-based digital twin of an experiment with a scale model ship," *Computer-Aided Design*, 103191, 2022.

[32] R. Stark, C. Fresemann, and K. Lindow, "Development and operation of digital twins for technical systems and services," *CIRP Annals*, pp. 129–132, 2019.

[33] K. Bruynseels, F. S. Sio, and J. V. D. Hoven, "Digital twins in health care: ethical implications of an emerging engineering paradigm," *Frontiers in Genetics*, vol. 9, no. 31, 2018.

[34] I. Graessler and A. Poehler, "Intelligent control of an assembly station by integration of a digital twin for employees into the decentralized control system," *Procedia Manufacturing*, vol. 24, pp. 185–189, 2018.

[35] B. Beate and H. Vera, "Digital twin as enabler for an innovative digital shopfloor management system in the ESB logistics learning factory at reutlingen – University," *Procedia Manufacturing*, vol. 9, pp. 198–205, 2019.

[36] C. Zhang, G. Zhou, J. He, Z. Li, and W. Cheng, "A data-and knowledge-driven framework for digital twin manufacturing cell," *Procedia CIRP*, vol. 83, pp. 345–350, 2019.

[37] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," I*EEE Computer Architecture Letters*, vol. 29, no. 12, pp. 1104–1113, 1980.