Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Three-Level Agent-Based Framework For Resource Allocation Optimization In Flexible Manufacturing Systems

Atef Gharbi[1,2*] , Sami Aziz Alshammari[3] , Mohamed Ayari[3,4] , Yamen El Touati[5,6]
Akil Elkamel[1] , Mahmoud Salaheldin Elsayed[5] , Zeineb Klai[5]

[1]*Department of Information Systems, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia, atef.gharbi@nbu.edu.sa*
[2]*LISI Laboratory, National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Carthage 1054, Tunisia*
[3]*Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia*
[4]*SYSCOM Laboratory, National Engineering School of Tunis, University of Tunis El-Manar, Tunis, Tunisia.*
[5]*Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia*
[6]*OASIS Laboratory, National Engineering School of Tunis, University of Tunis El Manar, Tunisia.*
E-mail: *atef.gharbi@nbu.edu.sa*

Agent-based approach has become a prominent paradigm for analysis, design and implementation of modern industrial systems. The paper focuses on the development of practical applications to optimize resource allocation in flexible manufacturing systems using auto-guided vehicles (AGVs). The system uses Agent Unified Model Language (AUML) to design models for various agents such as machines, vehicles, and products. In this paper, we describe in detail a multi-level agent-based framework for optimizing resource allocation in flexible manufacturing systems. In the first level, we define the role, capacity, and function of the agent in the system and describe its functions in detail. The second level deals with the behavior of these agents and explains in detail their decision-making processes and tasks. The third level examines how agents interact and communicate to achieve system-wide objectives. Using Contract Net Protocol (CNP), agents can demonstrate and manage interactions, ensuring efficient negotiations and resource allocation. The multi-level approach offers effective solutions for optimizing operations of dynamic and distributed manufacturing systems.

Keywords: Multi-Agent System (MAS), Auto-Guided Vehicles (AGVs), Flexible Manufacturing System, Autonomous Agents, Multi-Level Approach.

1. Introduction

A flexible manufacturing system (FMS) is used in modern industrial environments due to their increasing complexity, flexibility, and efficiency. They are designed to handle variations in product types and quantities, as well as changes in production requirements. Among the main components of this system are autonomous driving vehicles (AGVs) that move materials between production lines independently, reducing human intervention and improving workflow efficiency. Coordinating and optimizing AGVs with other systems, such as machines and products, is a challenging task. Conflict resolution, resource allocation, and scheduling are examples of these questions. Controlling these systems traditionally is not sufficient to cope with the dynamic and unpredictable nature of real manufacturing environments. As a result, multi-agent systems (MASs) have become more popular for modeling and simulating these environments. Multi-agent systems (MASs) have gained considerable attention in recent years due to their potential to improve efficiency, flexibility, and scalability in flexible manufacturing systems (FMSs). MAS allows agents such as vehicles, machines and products to communicate, collaborate and make autonomous decisions, leading to a more adaptive and robust production system. In complex production environments, Pilikottil et al. emphasize the importance of MAS in dynamic planning, resource allocation and decision-making. To respond to the growing complexity of modern manufacturing systems, they emphasize the need for better coordination mechanisms and communication protocols, and the integration of MAS with advanced technologies such as Internet of Things (IoT) and Artificial Intelligence (AI) [1]. Several studies have explored the integration of Reinforcement Learning (RL) into MAS frameworks to improve decision-making and optimization. Bahrpeyma and Reichelt comprehensively describe the Multi-Agent Reinforcement Learning (MARL) of smart factories, and demonstrate their effectiveness in solving complex planning, resource management and optimization problems in dynamic environments [2].

As Kim et al. showed in their study [11], MAS combined with RL can improve production flexibility and efficiency in smart manufacturing systems.

In FMS, dynamic resource allocation and scheduling are critical challenges. According to Bi et al., dynamic resource allocation requires real-time decision-making and flexibility in environments with variable production demands. MAS is used to optimize machine, vehicle, and other resource use, resulting in improved system efficiency [3]. According to Zhang et al., deep reinforcement learning (DRL) can be used to manage complex production workflows using MAS [10].

May et al. introduce a decentralized MAS for production control, where agents use economic models to bid on resources. The decentralized approach increases the flexibility and scalability of the FMS by allowing it to respond to fluctuations in production demands [4]. According to Egger et al. [6], a decentralized scheduling method can improve cooperative MAS in real-world manufacturing environments by increasing deployment ease and scalability.

In FMS, agent-based scheduling is a prominent application of MAS. In flexible manufacturing systems, Messinis and Vosniakos developed a Petri-net-based system that integrates MAS with deadlock avoidance strategies [9]. Popper et al. used multi-agent reinforcement learning to demonstrate that simultaneous production and AGV scheduling improve coordination between mobile agents and production processes, reducing delays and increasing efficiency [16]. According to Kovalenko et al., cooperative product agents are designed to increase the flexibility of manufacturing systems by enabling models-based decision-making. It improves the system's response to changes in production requirements and ensures optimal resource utilization [5]. Dittrich and Fohlmeister describe a co-operative MAS that uses reinforcement learning to improve collaboration between agents and leads to more efficient and adaptable production control [18].

According to Komesker et al., modular production systems can be resiliant against disruptions and changes in production flows by using MAS frameworks. It is crucial to maintain continuous operation in dynamic environments, especially in modular and reconfigurable manufacturing systems [8].

Johnson et al. show that MAS can manage real-time production changes and optimize production processes in highly dynamic manufacturing environments with MARL [14]. The graphical-based MAS has recently improved the coordination and decision-making capabilities of agents. Jing and al. developed a graph-convolutionary network-based MAS to improve communication and decision-making between agents in order to manage complex production workflows [15]. Yun et al. studied a multi-agent agent-based decision-making system for sustainable manufacturing based on explaining multi-agent deep reinforcement learning. [12]. Asgrawal et al. proposed a framework for integrating autonomous mobile robots (AMRs) into the MAS and showed how mobile agents can interact with stationary agents (such as machines) to optimize production flows. The integration of mobile and stationary agents increases the flexibility and scalability of manufacturing systems [17].

This paper presents a structured approach to the design and implementation of multiple agent systems (MASs). For clarity and comprehensiveness, the framework is composed of three main levels. At the first level, the agent itself is described, including role, attributes, and abilities. The second level explores the behavior of these agents and how they act independently to achieve their own goals. Finally, the third level examines the interaction between agents and emphasizes the cooperative and communication process that allows systems to work effectively.

Section 2 presents the first level, which gives an overview of MAS architecture, and defines the roles, abilities, and functions of different agents, including machine agents, AGV agents, and product agents. In section 3, we take a second level, which focuses on the behavior of agents. Section 4 explores the third level and explains in detail how agents interact and communicate with each other. Finally, in Section 5, we summarize the contributions proposed for a multilevel MAS framework.

2. First level: MAS with agent overview

In flexible manufacturing systems (FMS), vehicle, machine, product, and environment interactions play a critical role in improving efficiency and resource management. Within the system, each of these agents has specific roles and responsibilities:

1. Vehicle Agent: transports parts between machines in the FMS. It responds to transport requests from product agents. Vehicle agents decide based on availability and certain criteria, such as proximity to the product or load capacity. This ensures efficient transportation of parts between machines by transitioning between free, busy, and standby states.

2. Machine Agent: handles the processing of parts within the FMS. This agent reacts to requests from product agents to transform semi-finished parts into new forms. There is an entry stock where parts are waiting to be processed, and an exit stock where processed parts are stored if no vehicle is available for immediate transport. As a result, each part is processed in accordance with its availability and production schedule.

3. Product Agent: This is the most advanced agent of the system. In production, it represents the parts moving from one stage to another. In interaction with vehicles and machinery agents, product agents initiate transport and processing requests. It is responsible for delivering parts through the production line quickly and efficiently. Coordination with other agents, negotiations on resources and production plans are the main objectives of product agents.

4. Environment Agent: Coordinates the interactions between vehicles, machines and products. Maintaining a global system view ensures that all agents are aware of the state of other agents (e.g., machine and vehicle availability). The environment agent maintains a centralized resource status, coordinates resource allocation and scheduling, and facilitates real-time communication between agents, ensuring smooth system operation.

Dynamic interactions between these agents model the flexible manufacturing system. Transport is managed by the Vehicle agent, processing is controlled by the Machine agent, production is driven by the Product agent, and coordination is provided by the Environment agent. By optimizing production, allocating resources efficiently, and minimizing conflicts, the FMS is highly adaptable and efficient.

Our approach simulates flexible manufacturing systems by closely replicating real-world conditions. Since the case is both a simulation and a scheduling optimization, the environment, as well as the agents interacting with both the environment and each other, are crucial to the simulator. The following section introduces the most important classes for implementing the simulator.

Agent modeling is achieved by inheriting classes from the JADE platform's Agent class (Figure 1). Agents are assigned specific roles that dictate their behavior. Using the JADE platform, this agent-based simulation approach provides a flexible and realistic representation of AGV systems.

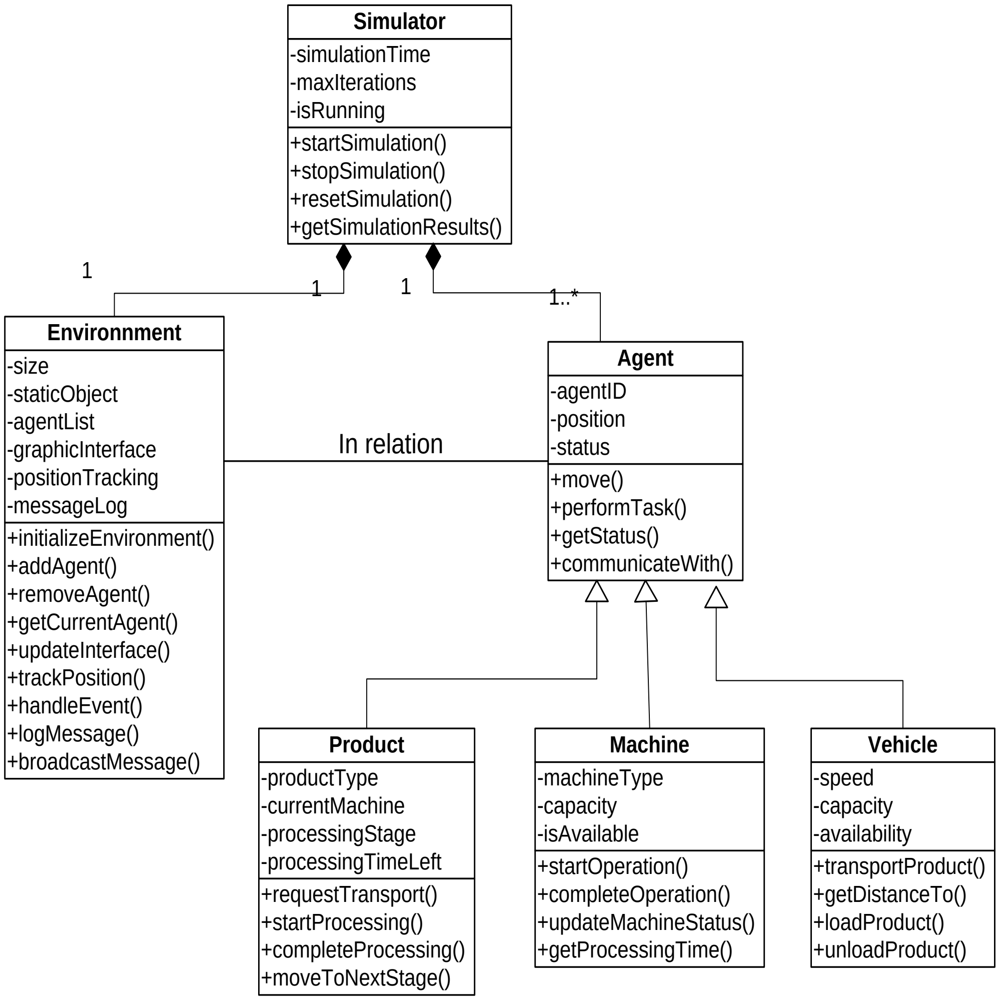

Figure 1. Class diagram Class Diagram for Agent-Based Simulation Environment

2.1 The environment agent

The environment agent serves as the dynamic interface for the simulation, facilitating the actions, decisions, cooperation, interactions, and mutual perception of the various agents. Environment agents consist of two types of objects structurally:

The static objects include the flow and input/output stocks for each machine, which are directly associated with the simulation of production in an industrial setting.

A dynamic object is a representation of a vehicle or machine that changes as it interacts or evolves with its surrounding environment.

Environment agents respond to actions initiated by actors (vehicles, machines) through a continuous exchange of messages. Here are the main functions of the environment agent:

- It provides a graphical interface for entering simulation parameters and visualizing the simulation's overall state in real-time.
- It can locate agents (track their positions) and react to direct and indirect stimuli.
- It collects information about the vehicles' positions and the machines' states.

To maintain a comprehensive view of the production system and ensure its efficient management, the environment agent functions as a central coordinator and information hub. Most of the messages exchanged between the agents are between the actors (vehicles, machines) and the environment. The agents frequently ask the environment for information about the current state of the system and the data they perceive. As a result, the agents have access to all the information they need for their actions and for analyzing data in the environment.

The environment agent stores the following key information during simulation:

- Vehicles, machines, and products, along with their identifiers and current conditions.
- The position of vehicles as well as the status of input and output stocks.
- The management strategy adopted for the system.

The environment agent, therefore, provides a global view of the simulation and facilitates real-time decision-making by coordinating and centralizing information.

2.2 The vehicle agent

The vehicle agent transports parts between machines within the system as a reactive agent. In simulations, it is primarily characterized by its speed, which affects its performance. Specific criteria are used to determine which vehicle is most appropriate for a particular task. Among these criteria are:

- Proximity: Priority is given to the vehicle nearest to the part. When multiple vehicles are equally close, the first to respond will be selected.

- Speed: The fastest vehicle ensures that parts are transported as quickly as possible between machines.

- Load: Use the lowest-load vehicle to avoid overloading and balance the workload of the agents.

Using this approach, the production system is more efficient as the most suitable vehicle is chosen based on real-time conditions and performance criteria.

2.3 The machine agent

In machine agents, one part is transformed into another over a specified processing time. Two types of stocks are associated with it: entry stocks and exit stocks.

All incoming parts that cannot be processed immediately are stored in the entry stock when the machine is occupied. Once the part has been processed, it will be temporarily stored in the exit stock if there are no vehicles available to transport it to the next stage.

By coordinating with vehicles for transportation, and handling pending parts through entry and exit stocks, this mechanism ensures that the machine continues to run efficiently.

2.4 The product agent

Product agents exhibit a variety of behaviors that facilitate their interactions with other components of the system. Specifically, it coordinates parts transport with vehicle agents. Product agents select optimal vehicles based on predefined criteria, such as proximity or speed, after receiving responses from vehicles. As soon as a part is awaiting processing, the product agent sends a request to the appropriate machine agent and awaits a response confirming availability. The process is repeated multiple times throughout the part's journey through the system, ensuring efficient coordination between vehicles and machines.

3. Second level: Behavior of agents

A flexible manufacturing simulator is implemented using a multi-agent system, in which vehicles, machines, and products are represented as agents. The product agents play a central and critical role in the simulation process. By making decisions, initiating requests, and coordinating transportation and processing needs, product agents drive the production flow. The vehicle and machine agents, on the other hand, serve as support agents, providing information to the product agents and completing their transport and processing requirements. This agent-based structure allows the product agents to control the workflow while ensuring that vehicle and machine resources are utilized efficiently to meet the demands of the system.

In a system, all agents operate within a shared environment with other agents. Both machine and vehicle agents perceive similar stimuli (i.e., messages from product agents) and act similarly to meet the needs of these agents. A vehicle can transport parts of a product if they are available, while a machine can process parts if they are available. Product agents, however, are the most sophisticated. The agents not only perceive messages from other agents in the system, including product agents, machines, and vehicles, but also negotiate with them. During these negotiations, their actions involve gathering information from the vehicle and machine agents.

Through interaction with each other, product agents strive to perform tasks quickly and efficiently.

Agent goals: Each agent has a set of goals based on the role of each agent in the system, including:

- Product Agents: The goal is to complete tasks as quickly as possible by communicating with machines and vehicles.
- Vehicle and Machine Agents: Their main goal is to provide transportation and processing support to product agents.

Table 1 illustrates the distinct characteristics and interactions of the agents within the multi-agent system.

Table 1. The characteristics of the agents in the system.

Agents	Perceptions	Actions	Goals	Environment
Vehicle	Receives messages from product agents	Transport parts between machines	Efficiently transport parts with minimal delays	Interacts with all other agents (products, machines, environment)
Machine	Receives messages from product agents	Process parts based on availability	Process parts efficiently to minimize idle time	interacts with all other agents (products, vehicles, environment)
Products	Receives messages from machine, vehicle, and other product agents	1. Wait for permission from higher-priority product agents 2. Request transport from vehicle agents and processing from machine agents	Complete production tasks as quickly as possible by optimizing resource usage	Interacts with all other agents (machines, vehicles, other products, environment)

The focus of this paper is on the social behaviors that arise from the interactions and coordination of agents. Since autonomous agents have no complete knowledge of their environment, the main challenge is to manage the dependencies between their activities. The constant evolution of the environment is complemented by unpredictable actions, reactions and objectives of the agents. To facilitate agent behavior, the JADE API provides several predefined behavior types, including

- CyclicBehaviour (CyclicB): Implement cyclic behavior that performs behavior in several instances.
- OneShotBehaviour (OneShotB): Executes a single action executed once.

Using these types of behaviours as the basis, we made specific choices for modeling agent behaviours, as shown in Table 2. These options help define how agents interact, perform tasks, adapt to changing environments, ensuring individual autonomy and effective coordination at the system's level.

Table 2. Agent behaviors

Agent	Behaviour	Type	Role
Product	CollectionResponse	CyclicB	Manages the sequence of part transfers between machines. The agent searches for an available vehicle and waits until the machine is ready for processing.
	ProductRank	OneShotB	Waits for a message indicating the product's rank in the production queue. Upon receipt, the **Permission** behavior is triggered.
	Permission	OneShotB	Waits for messages from higher-priority products before initiating the **CollectionResponse** behavior.
Vehicle	StatusUpdate	CyclicB	Cyclically sends status updates and coordinates to the environment, indicating its current availability and location.
	CoordinationV	CyclicB	Responds to product agent requests: sends "OK" if available, otherwise sends "REFUSED".
	ConflictResolution	CyclicB	Periodically checks for conflicts with other vehicles. If a conflict arises, prioritizes the vehicle further along in its task.
	Proposal	OneShotB	Proposes a schedule based on a selected metaheuristic and sends the cost to the environment. If chosen, the vehicle receives the order for execution.
	Termination	OneShotB	Terminates its operations when it receives a termination message from the environment.
Machine	StockUpdate	CyclicB	Periodically sends updates on its status, including incoming and outgoing stock, to the environment.
	Coordination M	CyclicB	Responds to product requests: sends "OK" if available for processing, otherwise sends "REFUSED".
	Termination	OneShotB	Terminates its operations when it receives a termination message from the environment.

Agent behavior can be represented using state-transition diagrams. They show finite state automata as state graphs connected by directed arcs, which represent state transitions.

Product agents change from one state to another depending on messages and events received. As shown in Figure 2, these states describe their lifecycle in the system:

1. Awaiting: This is the initial state of the product after its creation. The product must first request transportation from the vehicle agent before processing begins. The product remains in this waiting state until a positive response is received.

2. Part transport approval: Once the vehicle agent has approved the product, it is transferred to this state. Then, based on a pre-determined criteria (proximity, usage frequency, speed, etc.), the best options are selected and a confirmation is sent. The parts are then transported.
3. Standby Machine: After transporting a part, the processing is requested by the machine agent concerned. As long as the machine is not occupied, the product is temporarily placed in the input stock and awaits availability.
4. Final processing: Once the machine is available, the product is processed. After processing, the product either reaches the end of the production cycle (if all steps are completed) or enters a standby, repeating the cycle as needed, until it reaches the end of the entire production range.

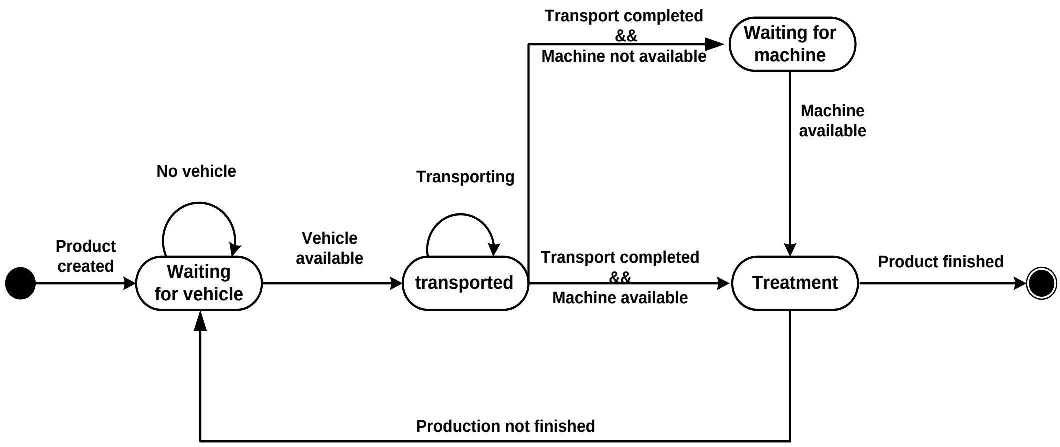

Figure 2. State diagram of the product Processing Workflow

4. Third level: Agent Interaction

In the Contract Net Protocol (CNP) and its related family of protocols, there is an Initiator (of which there is exactly one) and a Participant (of which there may be one or more). It is the initiator's responsibility to (i) initiate and orchestrate the negotiation process, (ii) gather, evaluate, and compare the proposals (bids) submitted by the Participants, and (iii) communicate the results to the Participants, concluding the negotiation.

We demonstrate the interaction dynamics between agents within our system using the Contract Net Protocol (CNP). Agents use this protocol to coordinate tasks and allocate resources, highlighting the negotiation and communication processes between them. By using CNP, we model the decentralized decision-making and collaborative behaviors characteristic of multi-agent systems.

4.1 Optimal scheduling calculation

It is possible for different products to compete for the same limited resources, resulting in conflict or even mutual blockage. A schedule of production tasks should be created before starting the simulation to mitigate this problem. Schedules for flexible manufacturing systems are static (offline) and are not affected by their dynamic characteristics. In fact, it takes into account factors such as the number of products and their production sequences, as well as the number of available machines. Multiple entities (agents) can collaborate to choose the optimal scheduling solution.

The environment agent plays an important role in this context by requesting the vehicle agents to propose a schedule that minimizes the total execution time. As a result, the vehicle agents determine a schedule with the lowest cost based on the optimization approach they have chosen. Upon completion of their evaluations, each vehicle agent sends its proposal to the environment agent.

The environment agent waits until all vehicle agents have responded. Once all proposals have been received, the environment agent selects the best schedule and requests data associated with the selected schedule from the vehicle agent. Once the request is received, the vehicle agent provides the necessary schedule information. The environment agent assigns each product agent a corresponding rank by using the schedule to ensure that the production process goes as planned. This reduces conflicts between scheduling and resource allocations and improves production efficiency.

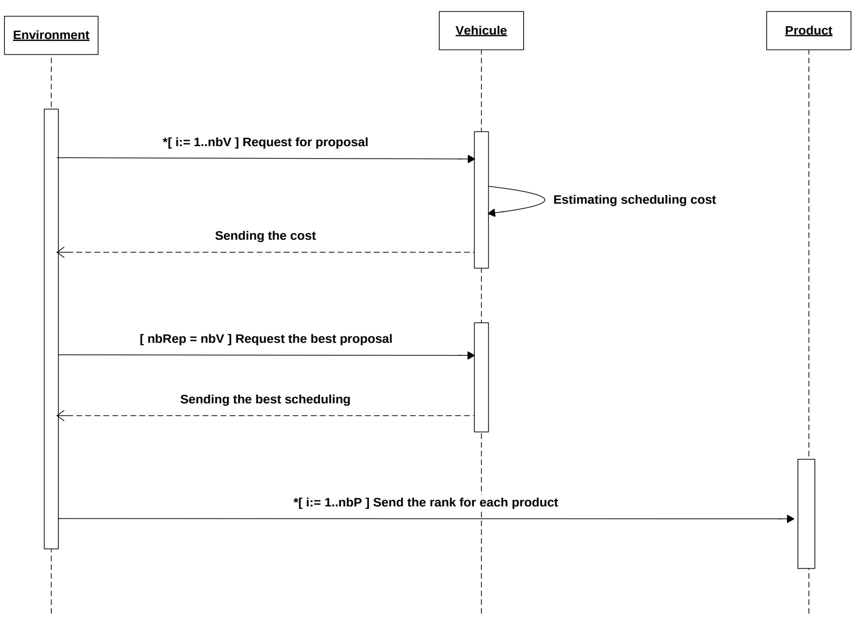

Figure 3. Sequence diagram of the computation of the optimal scheduling

4.2 Coordination between product agents and vehicle agents

Transporting parts between machines is handled by vehicle agents (Figure 4). The product agent initiates the transport by contacting all available vehicle agents. Vehicle agents evaluate their current availability after receiving the request: if they are unavailable, they refuse the request; otherwise, they propose to transport it. A product agent gathers all responses from the vehicle agents and evaluates them. Using predefined criteria (e.g., speed, proximity, or load), it selects the best proposition and sends an acceptance message to the chosen vehicle agent. After receiving the acceptance, the vehicle agent confirms the proposal and prepares to transport the part.

Using this process, the most appropriate vehicle agent is selected to transport the part efficiently, optimizing the flow of work within the system.

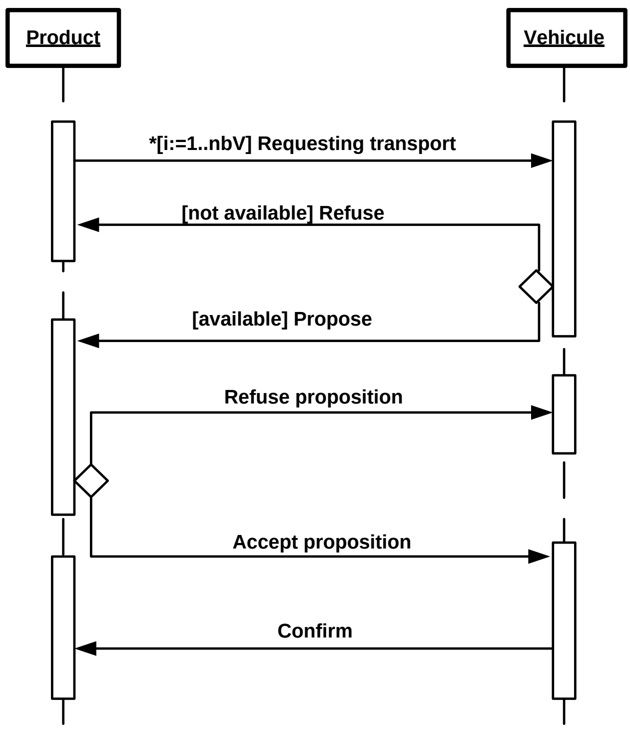

Figure 4. Diagram of cooperation sequences between the product and the vehicle agent

6. Conclusion

This paper describes a three-level approach to design and implementation of multi-agent systems (MAS). In the first level, we focused on defining the role, capabilities, and contributions of the agents to the system. The second level examines agent behavior especially how agents make decisions and perform tasks to achieve specific goals. At the third level, we investigated the interaction between agents, highlighting collaborative and communication processes that promote efficient allocation and coordination of resources. We demonstrate how the Contract Net Protocol (CNP) facilitates agents' interactions and ensures an efficient allocation of tasks through system structure. Based on multi-level analysis, this approach improves system flexibility and scalability, as well as decision-making and cooperation. It provides a solid basis for future developments in multi-agent systems, especially for optimizing complex dynamic environments such as flexible manufacturing systems.

References

[1] Pulikottil, T., Estrada-Jimenez, L. A., Rehman, H. U., Barata, J., Nikghadam-Hojjati, S., & Zarzycki, L. (2021, September). Multi-agent based manufacturing: current trends and challenges. In 2021 26th IEEE international conference on emerging technologies and factory automation (ETFA) (pp. 1-7). IEEE.
[2] Bahrpeyma, F., & Reichelt, D. (2022). A review of the applications of multi-agent reinforcement learning in smart factories. Frontiers in Robotics and AI, 9, 1027340.
[3] Bi, M., Kovalenko, I., Tilbury, D. M., & Barton, K. (2021). Dynamic resource allocation using multi-agent control for manufacturing systems. IFAC-PapersOnLine, 54(20), 488-494.
[4] May, M. C., Kiefer, L., Kuhnle, A., Stricker, N., & Lanza, G. (2021). Decentralized multi-agent production control through economic model bidding for matrix production systems. Procedia Cirp, 96, 3-8.
[5] Kovalenko, I., Balta, E. C., Tilbury, D. M., & Barton, K. (2022). Cooperative product agents to improve manufacturing system flexibility: A model-based decision framework. IEEE Transactions on Automation Science and Engineering, 20(1), 440-457.
[6] Egger, G., Chaltsev, D., Giusti, A., & Matt, D. T. (2020). A deployment-friendly decentralized scheduling approach for cooperative multi-agent systems in production systems. Procedia Manufacturing, 52, 127-132.
[7] Dornhöfer, M., Sack, S., Zenkert, J., & Fathi, M. (2020). Simulation of smart factory processes applying multi-agent-systems—a knowledge management perspective. Journal of Manufacturing and Materials Processing, 4(3), 89..
[8] Komesker, S., Motsch, W., Popper, J., Sidorenko, A., Wagner, A., & Ruskowski, M. (2022). Enabling a multi-agent system for resilient production flow in modular production systems. Procedia CIRP, 107, 991-998.
[9] Messinis, S., & Vosniakos, G. C. (2020). An agent-based flexible manufacturing system controller with Petri-net enabled algebraic deadlock avoidance. Reports in Mechanical Engineering, 1(1), 77-92.
[10] Zhang, Y., Zhu, H., Tang, D., Zhou, T., & Gui, Y. (2022). Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. Robotics and Computer-Integrated Manufacturing, 78, 102412.
[11] Kim, Y. G., Lee, S., Son, J., Bae, H., & Do Chung, B. (2020). Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. Journal of Manufacturing Systems, 57, 440-450.
[12] Yun, L., Wang, D., & Li, L. (2023). Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing. Applied Energy, 347, 121324.
[13] Lu, R., Li, Y. C., Li, Y., Jiang, J., & Ding, Y. (2020). Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. Applied Energy, 276, 115473.

[14] Johnson, D., Chen, G., & Lu, Y. (2022). Multi-agent reinforcement learning for real-time dynamic production scheduling in a robot assembly cell. IEEE Robotics and Automation Letters, 7(3), 7684-7691.
[15] Jing, X., Yao, X., Liu, M., & Zhou, J. (2024). Multi-agent reinforcement learning based on graph convolutional network for flexible job shop scheduling. Journal of Intelligent Manufacturing, 35(1), 75-93.
[16] Popper, J., Yfantis, V., & Ruskowski, M. (2021). Simultaneous production and agv scheduling using multi-agent deep reinforcement learning. Procedia Cirp, 104, 1523-1528.
[17] Agrawal, A., Won, S. J., Sharma, T., Deshpande, M., & McComb, C. (2021). A multi-agent reinforcement learning framework for intelligent manufacturing with autonomous mobile robots. Proceedings of the Design Society, 1, 161-170.
[18] Dittrich, M. A., & Fohlmeister, S. (2020). Cooperative multi-agent system for production control using reinforcement learning. CIRP Annals, 69(1), 389-392.