INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Software Agent Simulation Design on the Efficiency of Food Delivery

Shahrinaz Ismail [a], Salama A. Mostafa [b,*], Zirawani Baharum [c,*], Aldo Erianda [d], Mustafa Musa Jaber [e, f]
Mohammed Ahmed Jubair [g], M. Hasmil Adiya [h]

[a] School of Computing and Informatics, Albukhary International University, Jalan Tun Razak, 05200 Alor Setar, Kedah, Malaysia
[b] Center of Intelligent and Autonomous Systems (CIAS), Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Johor, Malaysia
[c] Malaysian Institute of Industrial Technology, Universiti Kuala Lumpur, Persiaran Sinaran Ilmu, 81750 Johor Bahru, Malaysia
[d] Department of Information Technology, Politeknik Negeri Padang, Sumatera Barat, Indonesia
[e] Department of Medical Instruments Engineering Techniques, Al-Turath University, Baghdad 10021, Iraq
[f] Department of Medical Instruments Engineering Techniques, Al-Farahidi University, Baghdad 10021, Iraq
[g] Department of Computer Technical Engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University, Al-Muthanna, Iraq
[h] Department of Informatic Engineering, Faculty of Computer Science, Institut Bisnis dan Teknologi Pelita Pekanbaru, Pekanbaru, Indonesia

*Corresponding author: *salama@uthm.edu.my; zirawani@unikl.edu.my*

***Abstract*—Food delivery services have gained popularity since the emergence of online food delivery. Since the recent pandemic, the demand for service has increased tremendously. Due to several factors that affect how much time additional riders spend on the road; food delivery companies have no control over the location or timing of the delivery riders. There is a need to study and understand the food delivery riders' efficiency to estimate the service system's capacity. The study can ensure that the capacity is sufficient based on the number of orders, which usually depends on the number of potential customers within a territory and the time each rider takes to deliver the orders successfully. This study is an opportunity to focus on the efficiency of the riders since there is not much work at the operational level of the food delivery structure. This study takes up the opportunity to design a software agent simulation on the efficiency of riders' operations in food service due to the lack of simulation to predict this perspective, which could be extended to efficiency prediction. The results presented in this paper are based on the system design phase using the Tropos methodology. At movement in the simulation, the graph of the efficiency is calculated. Upon crossing the threshold, it is considered that the rider agents have achieved the efficiency rate required for decision-making. The simulation's primary operations depend on frontline remotely mobile workers like food delivery riders. It can benefit relevant organizations in decision-making during strategic capacity planning.**

***Keywords*— Food delivery efficiency; software agent simulation; system design; Tropos methodology.**

Manuscript received 31 Oct. 2023; revised 21 Nov. 2023; accepted 22 Feb. 2023. Date of publication 31 Mar. 2024.

I. INTRODUCTION

There has been a sudden hike in demand for food delivery services in Malaysia since the recent Movement Control Order (MCO) situation due to the COVID-19 pandemic. The year before, a protest by approximately 200 FoodPanda riders on 30th September 2019 in Klang Valley and Kota Kinabalu had caused a bottleneck in delivery service in the affected areas. In both situations, the working riders had work overload, affecting the efficiency of the delivery service. It was unfair to the affected riders and the customers when the service was not performed as expected.

On the other hand, riders also need help with problems when customers cancel their orders between when they pick up the orders and before they arrive at the customers' location, costing waste. Food delivery companies have no control over the space and time of the delivery riders due to the multiple factors contributing to additional riders' time on the road, including incorrect or missing delivery information, loss of Internet connectivity during delivery, mechanical delays, and breakdowns, among others.

This research focuses on the bottlenecks and efficiency issues that could happen on the ground during the riders' delivery services operations. There is a need to predict the efficiency of the riders within certain case territories, which can be done through software agent simulation. The simulation is designed based on data from a preliminary study of a well-known food delivery service in Malaysia. The simulation design aims to guide the simulation system development, in which an environment of software agents is animated to reflect the natural environment while producing statistical graphical output to show the efficiency measurement. To achieve this aim, the research objective is to design a simulation to measure food delivery riders' efficiency in a customizable case setting. This paper presents the method of producing the simulation design using an agent-based modeling technique.

II. MATERIALS AND METHODS

This section summarizes the methods related to this study. It is divided into three subsections: service efficiency, software agent simulation, and Tropos methodology.

A. Service Efficiency

The concept of efficiency was popularized in the 1950s and can be expressed technically and allocatively in economics, physics, and other branches of science. In research by Jola-Sanchez et al. [1], efficiency is put forth as a ratio of output divided by input to determine the relative performance of a subject to be measured. Efficiency helps in implying the best possible allocation of resources (i.e., allocative efficiency), in which the resources are transformed using the best available technique or technology (i.e., technical efficiency). From the perspective of service operations management, the allocated resources represent the organization's capacity to provide the service. Resource capacity here is not limited to space and tools but also to the workforce or workers, in which the measuring units can be translated into energy and cost. Efficiency is often referred to as operational efficiency in the case of an overall operations management system. The organization can deliver its services in the most cost-effective manner possible while ensuring that the customers' satisfaction with the quality is met.

There are many ways to calculate business efficiency. The general way of calculating efficiency is by dividing the cost or energy of the actual output by the capacity, both in the same unit. In short, efficiency is the percentage of adequate capacity achieved [2]. This measurement will give a result in percentage form. Companies often calculate the efficiency rate of the current capacity and simulate the forecasted or predicted efficiency to know the required capacity as part of the operations' strategic planning.

In the e-commerce era, delivery time is crucial to customer satisfaction and retention. Dholakia and Zhao [3] found that timing plays a significant role in the relationship between the attributes and satisfaction of the online store. When the concern is on time, then efficiency is the challenge. Delayed delivery and inefficiency in delivery service beyond the usual practice (e.g., one-hour delivery) will hurt satisfaction regardless of road and weather conditions and the environment.

According to Liu et al. [4], 25 percent of Chinese customers were unhappy with a delayed delivery or incorrect product. Their research also showed that delivery significantly positively affects customer satisfaction. Variables in order fulfillment, especially on-time delivery, dominate the impact on overall customer evaluations and satisfaction. Delivery is essential for non-store businesses, including online retailers, where there is a temporary gap between order placement and delivery of the ordered products [3]. Therefore, timely delivery is vital to customer satisfaction and loyalty throughout the online food ordering industry.

Efficiency prediction is a common practice in operations management, primarily when it can provide feedback on the company's performance that will affect and lead to the company's profit and loss in terms to come. Efficiency is often measured for capacity planning, which involves workforce (including working time), space (or working area or territory), and machines or equipment. However, most efficiency predictions relating to operations and logistics must be more documented in research literature since their novelty is vital to a company's competitive advantage. Yet, related works on this are found on machine and equipment efficiency prediction using machine learning [5], multi-layer perceptron neural network [6], regression model [7], human reliability analysis [8], and healthcare services using machine learning [9]. Unfortunately, finding work related to business-related efficiency prediction is challenging, especially using the simulation approach, since they are often used for commercial value.

B. Software Agent Simulation

Using agent-based modeling and simulation, simulation can be modeled and visualized for efficiency prediction. It is an approach to modeling complex systems with interactive, autonomous software agents [10]. Software agents, or agents, are claimed to have behaviors, often described by simple rules and interactions with other agents that, in turn, affect their behaviors [11]. Social systems can be modeled to compose agents communicating with each other and influencing each other, learning from their experiences, and modifying their actions to fit their environment better [12]. Apart from complex systems, the simulation could also model time-dependent processes and more general forms of agent-based modeling, including models designed to automate or search [13].

According to Macal and North [11], there are three elements to a typical agent-based model: a set of agents, their behaviors, and their attributes; a set of agent relationships and interaction methods; and the environment of the agents. On the same note, agents are considered to have specific essential characteristics, such as being a self-contained, module, and uniquely identifiable individual; autonomous and self-directed; having a state that varies over time; and being social, having dynamic interactions with other agents that influence its behavior [14].

Agent-based modeling (ABM) is a powerful simulation modeling technique that has been used in several applications, including applications to real-world business problems [15]. A simple agent-based simulation, which models each person as an autonomous agent following the rules, can predict emerging collective behavior. In one of the examples of ABM application, a simulation of customer behavior in a theme park or supermarket was developed to improve adaptability in labor scheduling as the factors related to capacity and demand [16]. Another work was on the design of an ABM capable of predicting the possibility of future crimes [17].

A standard methodology was proposed as a guide in modeling software agent simulation. It is called Tropos, a method of agent-oriented software engineering (AOSE) that covers the entire software development process on agent systems [18], [19]. Tropos provides a method for engineers and researchers to "design multi-agent systems that can take advantage of the societal model throughout the design process" [18]. It is used in this research to design the simulation system. The methodology consists of four phases [19], as follows:

- Early requirements analysis: This phase requires researchers to analyze a problem by studying its organizational setting. The output is an organizational model that would include actors, their goals, and related interdependencies.
- Late requirements analysis: This phase reflects the system-to-be state (as it is referred to in the software engineering domain) described within the operational environment, along with related functions and qualities.
- Architectural design: This phase requires researchers to visualize the system's global architecture in terms of subsystems interconnected through data, control, and other possible dependencies.
- Detailed design: This phase determines where the behavior of each architectural component (defined in the previous phase) is described in further detail.

In some cases, the architectural design is not presented as it merely removes the soft goal from the late requirements analysis to make the system design more explicit, preparing it for the next phase. Since this paper presents the software agent simulation design using Tropos methodology, the following two sections will discuss the analysis and design processes that adopt the early requirements analysis, late requirements analysis, and detailed design without the architectural design.

C. *Case Study*

The research case setting is based on a preliminary study of Malaysia's two popular food delivery services, FoodPanda and GrabFood. Two interviews were conducted during the initial research to understand the food rider delivery process flow. Overall, it is understood that the riders are restricted within a territory or area for each delivery shift for FoodPanda. In contrast, there is no boundary regarding GrabFood's delivery territory. For simulation design, the software agent environment will be confined within a territory, in which the rider agents will move within the simulation interface that represents a territory.

Variables for the simulation used in this research are derived from the preliminary and existing studies [20-30]. Three main stakeholders or actors are identified from the riders' point of view, which they interact with during their operations. These three actors are involved in the simulation proposed in this paper to represent the actors in the agent world: rider (i.e., the person who delivers food), customer (i.e., the person who makes the order), and vendor (i.e., the food outlet or restaurant that prepares the food ordered by the customer).

There are four main tasks that a rider performs, which depend on the order made by the customer and the vendor's readiness to prepare the order. In designing the simulation, this research adopts the Tropos methodology, in which the process starts with determining soft goals, challenging goals, and tasks for each actor. Table 1 is tabulated to show the soft goals, challenging goals, and functions for the three identified actors for this research: rider, vendor, and customer.

From the information in Table I, the simulation design is produced in phases, as discussed in the next section. Each actor has at least one qualitative soft goal, which can be achieved by at least one challenging goal. The challenging goal is more explicit than the soft goal and is supported by several tasks to achieve it. Therefore, each actor will need to perform some tasks to achieve their original soft goal, and these tasks can be animated or simulated in the agent simulation environment.

TABLE I
OFT GOALS, HARD GOALS, AND TASKS FOR RIDER, VENDOR, AND CUSTOMER

Actor	Soft Goal	Hard Goal	Task
Rider	Efficient food delivery	Pick up order	Accept job Go to Vendor Pick up food
		Deliver order	Find destination Contact Customer Deliver food Receive cash payment Receive cancel notification
Vendor	Fulfill customer's needs	Prepare order	Receive order Prepare food Pass food to the rider Cancel order
Customer	Delivery service satisfaction	Receive order	Set location Make order Submit Order Pay orders online Pay order via cash Cancel order Receive cancel notification

The leading actor in this research is the rider, with a specific soft goal of being efficient in food delivery, as shown in Table 1. The rider needs to achieve two main objectives: picking up the order from the vendor and delivering the order to the customer. Based on the process flow a rider goes through in the real world. The tasks that the rider needs to perform are accepting the job, going to the vendor, picking up the order, finding the destination to the customer's location, contacting the customer if there is any doubt or problem during the delivery, and delivering the food, receiving cash payment (if applicable), and receiving the order notification of cancellation (if the customer cancels the order after the rider has picked up the food from the vendor). To visualize this scenario, the Vendor and Customer need to be illustrated in the simulation setting. Hence, they are considered actors in this research setting, too.

III. Results and Discussion

A. *Results of Early and Late Requirements Analysis*

Based on the details in Table I, Tropo's early and late requirements analysis diagrams are produced. In this phase, the stakeholders are represented as actors who depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. The results are presented in a requirements specification that describes all functional and non-functional requirements for the system-to-be.

According to Tropos methodology, each actor and its environment of goals and tasks is illustrated in nodal form, i.e., circles. Each soft goal is described in a cloud symbol, each challenging goal is illustrated in a rounded rectangle, and each task is drawn in an elongated hexagon. Each symbol is connected with other symbols that relate to it by an arrow. There are two different types of arrows used in Tropos diagrams. One with a solid black arrowhead in the middle of its line shows explicit tasks to achieve the explicit challenging goal, while another arrow with a hairline head at the end of its line shows the link between a challenging goal and a soft goal.

Figure 1 shows the early requirements analysis diagram derived from the hard goals, soft goals, and tasks of the three actors presented in Table 1, namely rider, vendor, and customer. A circular symbol indicates each actor, with a bigger circle encapsulating every soft goal, hard goal, and task that belongs to that actor. Within the big circle, the goals are broken down into discrete tasks that could further clarify the tasks of the agents involved in the simulation.

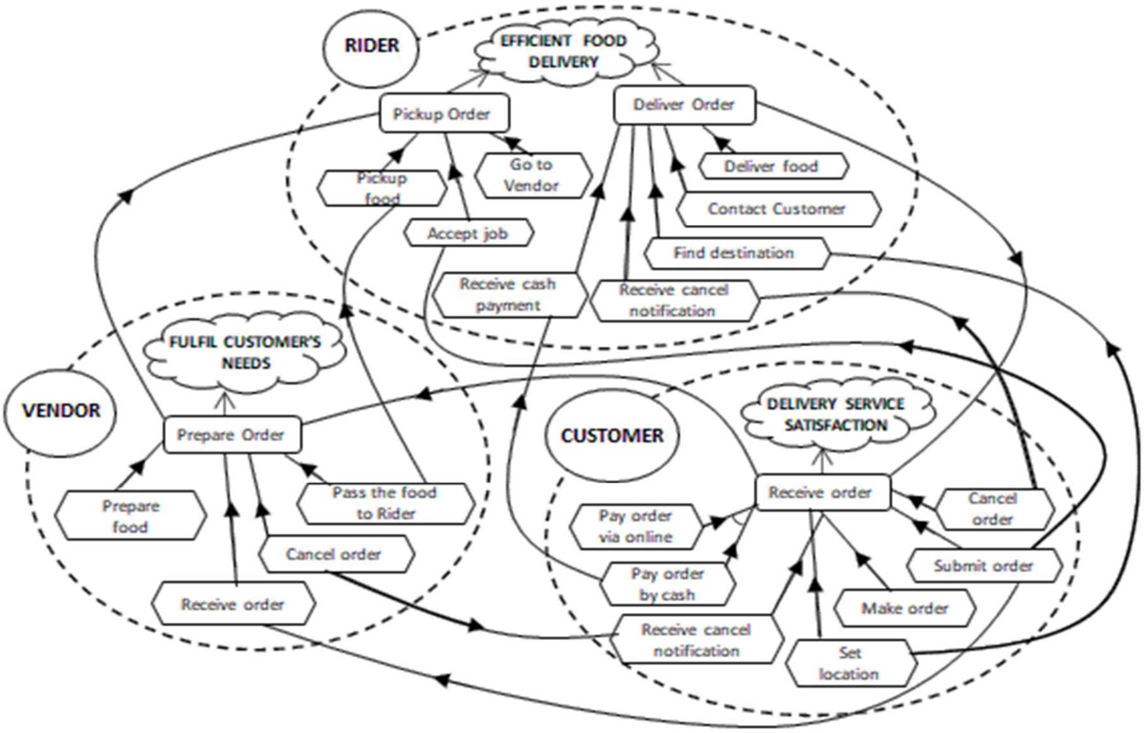

Fig. 1 Early requirements analysis diagram for the simulation

Figure 2 illustrates the late requirements analysis diagram for the simulation, derived after further analysis of the early requirements analysis diagram. Here, the simulation system is slowly introduced, which is the result of consolidating the challenging goals that could be automated or performed by an agent simulation. As shown in the diagram, the soft goal of the simulation system called a-Ride is to measure delivery efficiency. This soft goal is derived from achieving two challenging goals: calculating efficiency in minutes (time) and the number of successful deliveries (unit or order). The actors being detected here are the variables that will be part of the simulation interface, and the connecting lines show how they interact with the a-ride system.

To achieve the challenging goal of calculating efficiency in minutes, the simulation system needs to detect new orders made by customers and, at the same time, detect the nearest rider (i.e., agent), in which the distance between the nearest rider and the vendor that the customer orders the food from is closest found in the agent environment. Then, the system needs to calculate the rider agent's journey duration throughout one delivery service. On another part, to achieve the challenging goal of calculating efficiency in successful delivery, the simulation system also needs to be based on detecting new orders along with detecting canceled orders, then calculating the successful delivery out of the total number of orders made by customers.

The unit of efficiency is in percentage. The actual output and effective capacity units should be the same, i.e., if the actual production is in time (e.g., minutes), then the adequate capacity should be in minutes. The same goes for the unit of the number of orders, in which the actual output would be in the number of successfully delivered orders (minus the canceled orders and deliveries that never arrived in customers' hands). At the same time, adequate capacity is the most optimistic number of deliveries riders could achieve.

B. *Results of a-Ride System*

Based on the soft goal of the a-ride simulation shown in Figure 2, the two challenging goals are further broken down into manageable tasks presented in the system's detailed design, as shown in Figure 3. In short, the required resources for the simulation are notifications on accepted jobs by riders, confirmed orders, canceled orders, and some variables for calculation like time and number of successful deliveries.

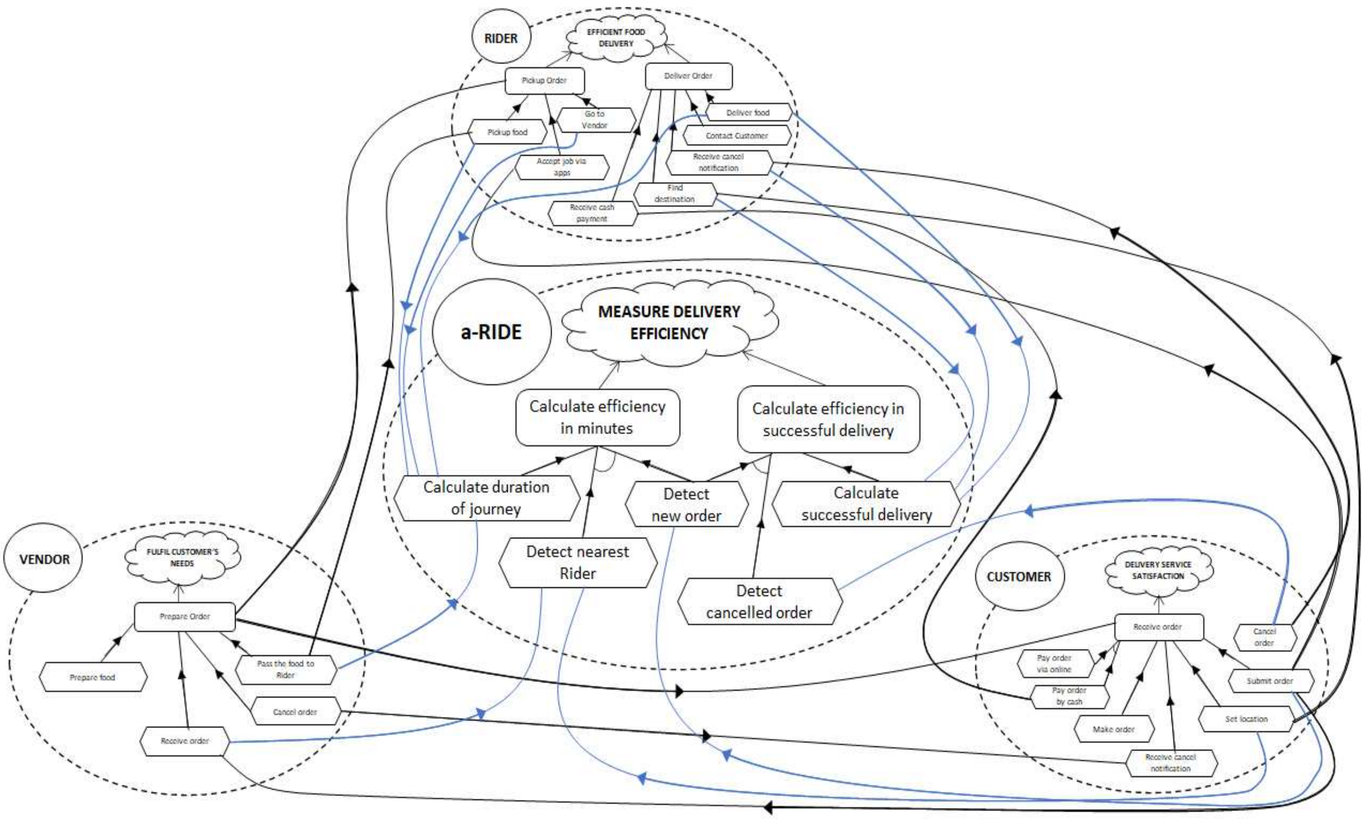

Fig. 2 Example of late requirements analysis diagram for the simulation

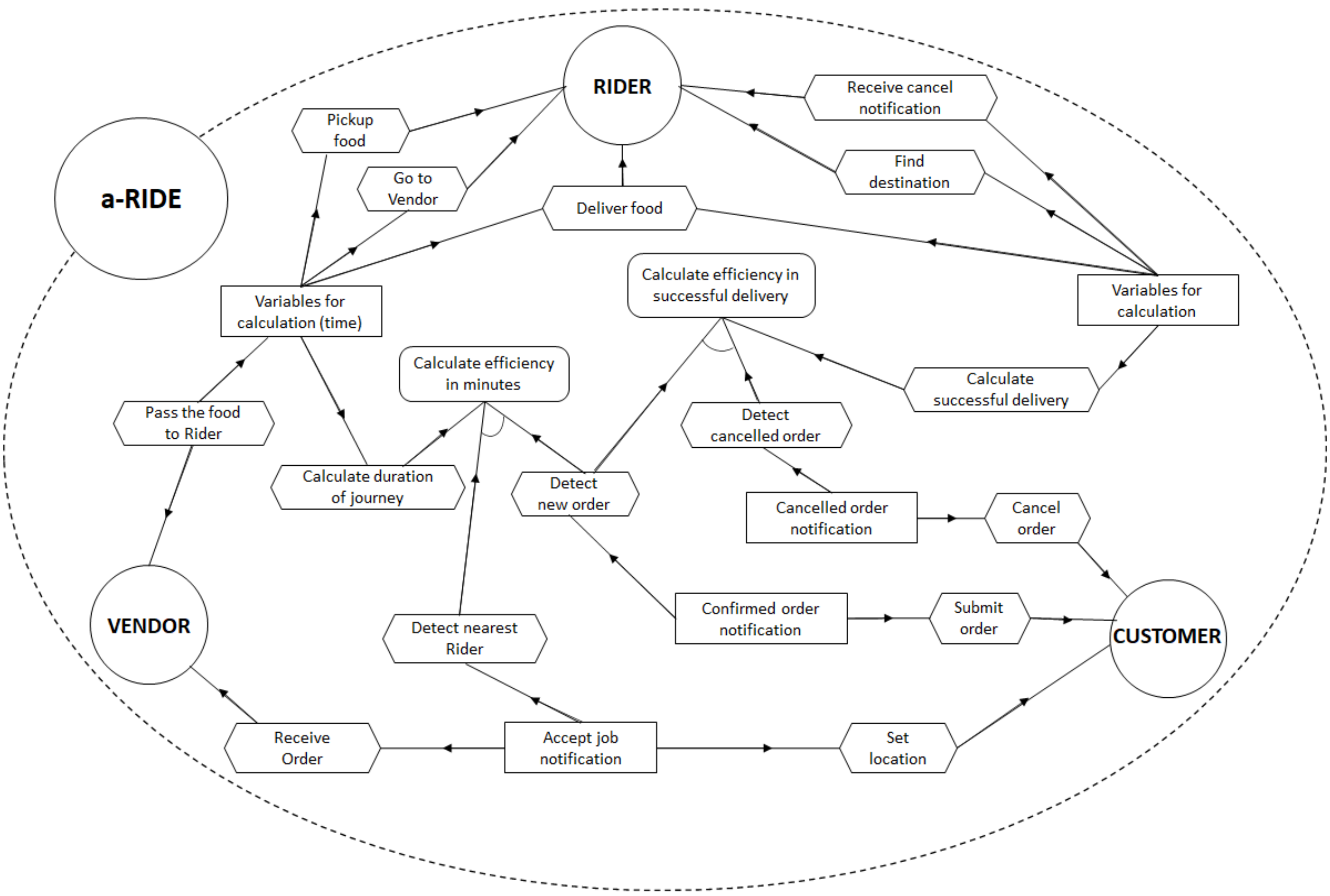

Fig. 3 System design for a-Ride simulation

Figure 3 shows that the original three actors are now part of the a-ride simulation (shown here, the big circle consists of all three actors - rider, vendor, and customer). This outcome means the simulation must show how these three interact in the agent environment. A new symbol, the rectangle, is introduced in Figure 3. These rectangles indicate resources, information, files, or variables that the actors refer to or act upon. For this case, the resources are variables for both calculations (as discussed above): confirmed order notification, canceled order notification, and accept job notification. These resources also trigger the algorithm to pick up behind the scenes of the simulation interface.

The simulation system interface design is proposed with the planning done, as shown in Figure 4. The proposed simulation is called a-Ride and is designed using a multi-agent programmable modeling environment software called NetLogo. As shown in Figure 4, the interface consists of three sliders, two buttons, one toggle button, and one graph view. The first slider is called num_rider, allowing users to set the number of riders they want to simulate. The second slider is called num_customer, which enables the user to set the number of Customers to simulate. The efficiency_threshold slider is used to set the threshold of efficiency that the user wants to simulate, commonly set at 60 percent up to 90 percent for a range of expected efficiency. The on-off toggle button called canceled_order sets whether the simulation should take into account canceled orders or not.

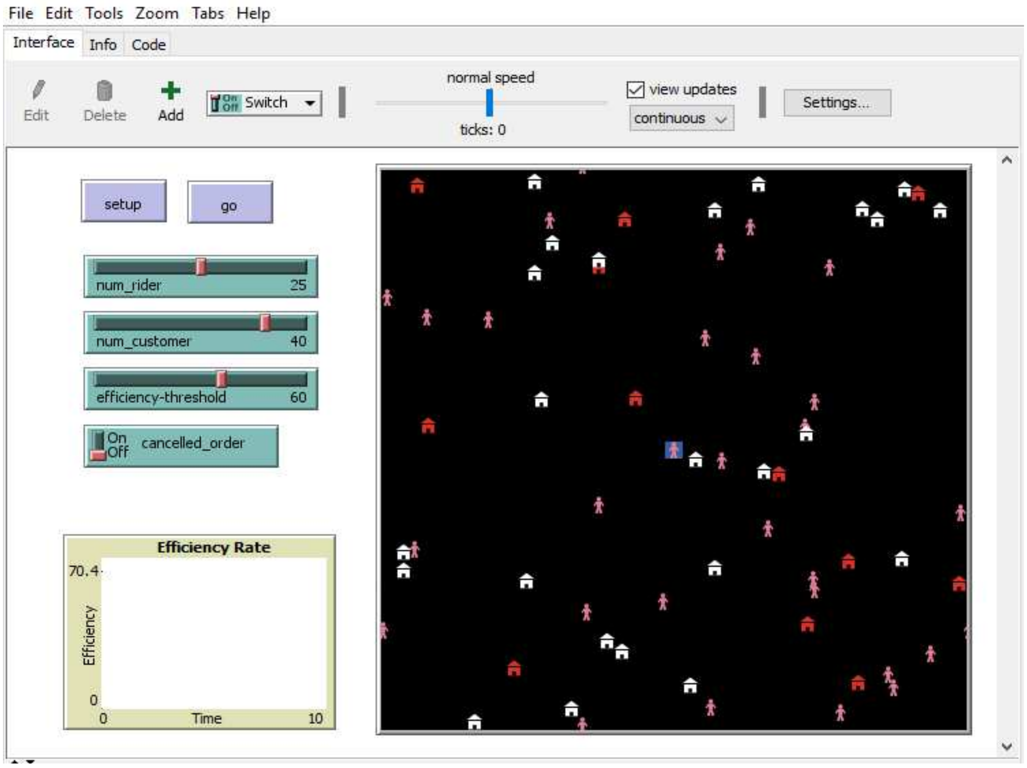

Fig. 4 Interface design of a-ride simulation

The setup and go buttons are common in the NetLogo simulation interface. The setup button is used after choosing the options on the sliders and toggle button, while the go button is to be clicked after the agent environment is set up and to start the simulation with the settings. As the simulation runs, the graph will plot three lines: one horizontal line of the efficiency threshold set by the user, one line derived from the calculation of efficiency in minutes, and one line based on the calculation of the number of successful deliveries.

The agent world will show two types of turtles (a name representing agents in Netlogo): the rider shown as the person symbol and the customer shown as a house symbol. The blue pixelated box in the middle of the world represents a Vendor. Only one vendor is designed in this interface to reduce the complexity of programming.

The position of the houses and rider agents at the beginning of the simulation will be randomly placed. The house's color will be white by default and will turn red randomly to indicate that a customer has made a new order. Upon the color change, the nearest rider agent will be identified, provided that the agent is not in the middle of "delivering food" to other customers. The closest rider identified will then move towards the blue box, i.e., vendor, and move towards the red house, i.e., customer with the new order. During this animated journey, the code behind the scenes will calculate the duration of the trip. Successful delivery upon the rider agent's arrival at the customer's red house and turning the house color green will be counted as one. If the cancelled_order toggle button is switched on, then a notification of canceled order will appear on the red house, and the house's color will turn white. This result will affect the movement of the rider agent as it will halt at its position as the notification appears, and then its status will be ready to receive a new order. As for the graph view, once the efficiency_threshold is set, the red horizontal line will appear to indicate the position of the threshold, as shown in Figure 5.

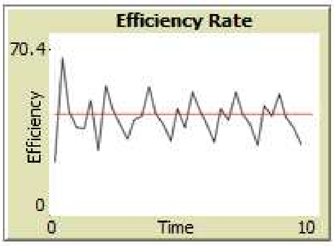

Fig. 5 Sample of the graph views

At every 'tick' of movement in the simulation, the graph will plot the graph dot by dot to form a black line due to the efficiency calculation. Upon crossing the red line, i.e., threshold, it is considered that the rider agents have achieved the efficiency rate required by the setting. However, the black line may not always be above the red line for long, as the efficiency rate may change as time passes, hence the graphical pattern, as shown in Figure 6, resulting from multiple runs.

The idea proposed in this paper is to simulate the riders in an agent environment to measure the efficiency rate when the number of riders is set differently during each simulation experiment. This simulation system is proposed to be an interface for users to measure and analyze the possibilities of efficiency rates based on a few factors, like the number of riders, customers, and the possibility of canceled orders. For example, suppose the number of riders is less. In that case, the number of potential customers in a territory is significant, but the efficiency rate expected from them is high, e.g., more than 60 percent. Can this be achieved? In another example, what if the number of riders and potential customers is also significant, but there are possibilities of canceled orders? How would the efficiency rate be? Users can play around with different scenarios like these two examples.

Since the simulation in an agent environment could run in a very speedy mode, i.e., at a default rate of 30 frames per second for an average speed, it would be challenging to capture the number of successful deliveries when the number of houses or customers is significant. The same goes for the movement of the rider agents, which will be difficult to observe when there are many running around in the agent environment. There is a way to zoom in on one rider agent for close observation, and this will require the primary platform to be ready first. Complex programming will be needed as the features of this simulation are added as the requirements are perceived.

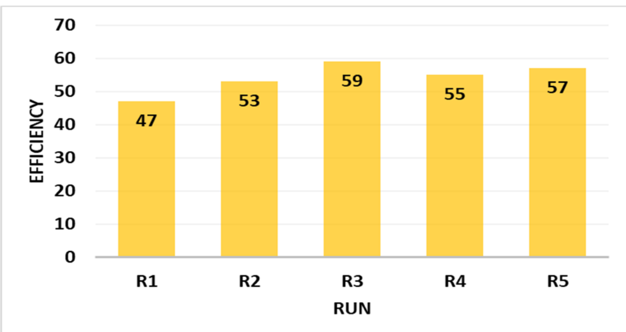

Fig. 6 Average efficiency rate of multiple runs

The simulation alone cannot make efficiency predictions; it merely measures the efficiency rate according to the settings done by the users. The prediction is based on how the users use this simulation system and how the settings are done. Having said this, the users need to note each set they make for each simulation to keep track of the results since this system does not store any simulation results every time it runs.

Nevertheless, it is perceived that this simulation will benefit organizations in the decision-making process during strategic capacity planning, in which their primary operations depend on frontline remotely mobile workers like food delivery riders. The a-ride tool measures efficiency rate to facilitate companies with similar working environments as food delivery services, like many other logistics and delivery services in the country. Future work will look into the development of this a-Ride simulation.

IV. Conclusions

Due to the various circumstances that contribute to additional riders' time on the road—such as inaccurate or missing delivery information, loss of Internet connectivity during delivery, mechanical delays, and breakdowns, among others—food delivery companies have no control over the location and timing of the delivery riders. This study takes up the opportunity to design a software agent simulation to investigate the efficiency of riders' operations in food service. The case study conducted in this research is based on a preliminary analysis of two popular food delivery services in Malaysia: FoodPanda and GrabFood. Overall, this research proves that software agent simulation can be used to measure the workforce's efficiency in an environment reflecting the real case scenario to help plan the capacity of employees' performance. The Tropos methodology was used during the system design phase to produce the results reported in this study. In the simulation, the efficiency graph is computed for each movement. The rider agents are deemed to have reached the efficiency rate necessary for decision-making once they cross the threshold. Food delivery riders and other frontline remotely mobile personnel are essential to the primary operations of the simulation. It can help pertinent companies make decisions during the strategic capacity planning process. Future work will include the full development of this agent simulation and a more complex view of the simulation, with the potential of predicting efficiency.

Acknowledgment

The Research and Innovation, Universiti Kuala Lumpur, funds this paper.

References

[1] A. F. Jola-Sanchez, A. J. Pedrazza-Martinez, K. M. Bretthauer and R. A. Britto. "Effect of armed conflicts on humanitarian operations: Total factor productivity and efficiency of rural hospitals", Journal of Operations Management, Vol. 45, pp. 73-85, 2016.

[2] R. Md Zani and M. R. M. Khir, Operations Management, 1st ed., Ch. 10, pp. 223-259, 2016. Published by Oxford Fajar: Shah Alam, Malaysia.

[3] R. R. Dholakia and M. Zhao, "Effects of online store attributes on customer satisfaction and repurchase intentions", International Journal of Retail & Distribution Management, Vol. 38, No. 7, pp. 482-496, 2010.

[4] X. Liu, M. He, F. Gao and P. Xie. "An empirical study of online shopping customer satisfaction in China: a holistic perspective", International Journal of Retail & Distribution Management, Vol. 36, No. 11, pp. 919-940, 2008.

[5] H. Sayadi, N. Patel, A. Sasan, H. Homayoun, "Machine Learning-based Approaches for Energy-Efficiency Prediction and Scheduling in Composite Cores Architectures", in Proceedings of IEEE 35th International Conference on Computer Design, pp. 129-136, 2017.

[6] A. F. Mashaly and A.A. Alazba, "MLP and MLR models for instantaneous thermal efficiency prediction of solar still under hyper-arid environment", Computers and Electronics in Agriculture, Vol. 122, pp. 146-155, 2016.

[7] M. Tomoskozi, P. Seeling, P. Ekler and F. H.P. Fitzek, "Regression Model Building and Efficiency Prediction of RoHCv2 Compressor Implementations for VoIP", in 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2016.

[8] E. Zarei, I. Mohammadfam, M. M. Aliabadi, A. Jamshidi, and F. Ghasemia, "Efficiency Prediction of Control Room Operators Based on Human Reliability Analysis and Dynamic Decision-Making Style in the Process Industry", American Institute of Chemical Engineers, Vol. 35, No. 2, pp. 192-199, 2015.

[9] A. Abdelaziz, M. Elhoseny, A. S. Salama and A.M. Riad, "A Machine Learning Model for Improving Healthcare services on Cloud Computing Environment", Measurement, Vol. 119, pp. 117-128, 2018.

[10] S. A. Mostafa, M. S. Ahmad, A. Ahmad, M. Annamalai, and S. S. Gunasekaran, "A Flexible Human-Agent Interaction model for supervised autonomous systems," In 2016 2nd International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR) (pp. 106-111). IEEE, 2016.

[11] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation", Journal of Simulation, Vol. 4, pp. 151–162, 2010. Published by Springer, Berlin, Heidelberg.

[12] S. A. Mostafa, S. S. Gunasekaran, M. S. Ahmad, A. Ahmad, M. Annamalai, and A. Mustapha, "Defining tasks and actions complexity-levels via their deliberation intensity measures in the layered adjustable autonomy model," In 2014 International Conference on Intelligent Environments (pp. 52-55). IEEE, 2014.

[13] S. A. Mostafa, M. S. Ahmad, M. Annamalai, A. Ahmad, and S. S. Gunasekaran, "Formulating dynamic agents' operational state via situation awareness assessment," In Advances in Intelligent Informatics (pp. 545-556). Springer, Cham, 2015

[14] S. Olariu and A. Y. Zomaya, Handbook of Bioinspired Algorithms, p. 679, 2006, Published by Chapman & Hall/CRC: Florida, USA.

[15] E. Bonabeau. "Agent-based modeling: Methods and techniques for simulating human systems", in Proceedings of the National Academy of Sciences (PNAS), 2002, pp. 7280-7287.

[16] E. Bonabeau. In G. Ballot & G. Weisbuch (Eds.). Application of Simulation to Social Sciences, Hermès Sciences: Paris, 2000, pp. 451-461.

[17] R.R. Brüngger, C. Kadar and I.P. Cvijikj, "Design of an agent-based model to predict crime (WIP)", in Proceedings of the Summer Computer Simulation Conference (SCSC '16), pp. 1-6, Article 55, 2016.

[18] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos and A. Perini, "Tropos: An agent-oriented software development methodology", AAMAS Journal, pp. 203–236, Vol. 8, No. 3, 2004.

[19] H.L. Zhang, C. Pang, X. Li, B. Shen and Y. Jiang "A Topological Description Language for Agent Networks", in Sheng Q.Z., Wang G., Jensen C.S., Xu G. (eds) Web Technologies and Applications. APWeb 2012, in Lecture Notes in Computer Science, pp. 759-766, Vol. 7235, 2012, Published by Springer, Berlin, Heidelberg.

[20] S. Ismail and M.S. Ahmad, "A goal-based framework on contextual requirements modelling for agent-mediated continual quality improvement (aCQI) in curriculum design", in Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (IMCOM '15), pp. 1–8, Article 16, 2015, Published by Association for Computing Machinery, New York, NY, USA, DOI: 10.1145/2701126.2701178.

[21] G. Zou, M. Gao, J. Tang, and L. Yilmaz. "Simulation of online food ordering delivery strategies using multi-agent system models." Journal of Simulation, Vol. 17, no. 3, pp.297-311, 2023.

[22] G. Zou, J. Tang, L Yilmaz, and X. Kong. "Online food ordering delivery strategies based on deep reinforcement learning." Applied Intelligence, pp. 1-13, 2022.

[23] G. Pezzotta, A. Rondini, F. Pirola, and R. Pinto. "Evaluation of discrete event simulation software to design and assess service delivery processes." Service Supply Chain Systems: A Systems Engineering Approach, vol. 8, no. 86, pp. 83-100, 2016.

[24] W. J. Chin, M. J. C. E. Lim, A. A. M. Yong, Al-Talib, and K. H. Chaw. "Service time performance analysis of improved automated restaurant by layout reconfiguration and conveyor system." In IOP Conference Series: Materials Science and Engineering, vol. 692, no. 1, p. 012003. IOP Publishing, 2019.

[25] E. Mangina, and I. P. Vlachos. "The changing role of information technology in food and beverage logistics management: beverage network optimisation using intelligent agent technology." Journal of food engineering, vol. 70, no. 3, pp. 403-420, 2005.

[26] S. Vongbunyong, S. P. Tripathi, K. Thamrongaphichartkul, N. Worrasittichai, A. Takutruea, and T. Prayongrak. "Simulation of autonomous mobile robot system for food delivery in In-patient ward with unity." In 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), pp. 1-6. IEEE, 2020.

[27] H. Fotouhi, N. Mori, E. Miller-Hooks, V. Sokolov, and S. Sahasrabudhe. "Assessing the effects of limited curbside pickup capacity in meal delivery operations for increased safety during a pandemic", Transportation Research Record, vol. 2675, no. 5, pp. 436-452, 2021.

[28] S. Abahussein, D. Ye, C. Zhu, Z. Cheng, U. Siddique, and S. Shen. "Multi-Agent Reinforcement Learning for Online Food Delivery with Location Privacy Preservation." Information, vol. 14, no. 11, p. 597, 2021.

[29] G. I. Fragapane, C. Zhang, F. Sgarbossa, and J. O. Strandhagen. "An agent-based simulation approach to model hospital logistics", International Journal of Simulation Modelling, vol. 18, no.4, pp. 654-665, 2019.

[30] D. J. McClements. "The future of food colloids: Next-generation nanoparticle delivery systems," Current Opinion in Colloid & Interface Science, vol. 28, pp. 7-14, 2017.